

普通高等教育“十一五”规划教材
21世纪大学数学创新教材
◎丛书主编 陈 化

数学建模与数学实验

汪晓银 周保平 主编



科学出版社
www.sciencep.com

普通高等教育"十一五"规划教材

21世纪大学数学创新教材

丛书主编 **陈 化**

数学建模与数学实验

汪晓银 周保平 主编



科学出版社

北京

内 容 简 介

本书通过实例与算法程序设计介绍了常用的数学建模方法，包括多元统计、时间序列分析、线性与非线性规划、多目标规划与目标规划、图论、动态规划、排队论、智能优化算法、微分与差分、模糊数学、神经网络、计算机仿真、灰色系统和层次分析法. 全书将建模技术与数学实验融为一体，注重数学建模思想介绍，重视数学软件(SAS、MATLAB、LINGO)在实际问题中的应用. 全书案例丰富，通俗易懂，便于自学.

本书既可以作为高等学校数学建模与数学实验课程的教材，也可作为本科生、研究生数学建模竞赛的培训教材或参考书籍，也是科学研究人员一本有价值的参考书籍.

图书在版编目(CIP)数据

数学建模与数学实验/汪晓银，周保平主编. 一北京：科学出版社，2010. 2

普通高等教育“十一五”规划教材. 21世纪大学数学创新教材

ISBN 978-7-03-026502-9

Ⅰ. 数… Ⅱ. ①汪…②周… Ⅲ. ①数学模型一高等学校一教材 ②高等数学一实验一高等学校一教材 Ⅳ. ①O141.4 ②O13－33

中国版本图书馆CIP数据核字(2010)第015571号

责任编辑：王雨舸 曾 莉/责任校对：董艳辉

责任印制：彭 超/封面设计：苏 波

科学出版社 出版

北京东黄城根北街16号

邮政编码：100717

http://www.sciencep.com

武汉市新华印刷有限责任公司印刷

科学出版社发行 各地新华书店经销

*

2010年1月第 一 版 开本：B5(720×1000)

2010年1月第一次印刷 印张：17 1/2

印数：1—5 000 字数：338 000

定价：28.00元

(如有印装质量问题，我社负责调换)

《21 世纪大学数学创新教材》丛书编委会



《21世纪大学数学创新教材》丛书序

《21世纪大学数学创新教材》为大学本科数学系列教材，大致划分为公共数学类、专业数学类两大块，创新是其主要特色和要求. 经组编委员会审定，列选科学出版社普通高等教育“十一五”规划教材.

一、组编机构

《21世纪大学数学创新教材》丛书由多所985和211大学联合组编：

二、教材特色

创新是本套教材的主要特色和要求，创造双重品牌：

先进. 把握教改、课改动态和学科发展前沿，学科、课程的先进理念、知识和方法原则上都要写进教材或体现在教材结构及内容中.

知识与方法创新. 重点教材、高层次教材，应体现知识、方法、结构、内容等方面的创新，有所建树，有所创造，有所贡献.

教学实践创新. 教材适用，教师好教，学生好学，是教材的基本标准. 应紧跟和引领教学实践，在教学方法、教材结构、知识组织、详略把握、内容安排上有独到之处.

继承与创新. 创新须与继承相结合，是继承基础上的创新；创新需转变为参编者、授课者的思想和行为，避免文化冲突.

三、指导思想

遵循国家教育部高等学校数学与统计学教学指导委员会关于课程教学的基本要求，力求教材体系完整，结构严谨，层次分明，深入浅出，循序渐进，阐述精炼，富有启发性，让学生打下坚实的理论基础. 除上述一般性要求外，还应具备下列特点：

(1) 恰当融入现代数学的新思想、新观点、新结果，使学生有较新的学术视野.

(2) 体现现代数学创新思维，着力培养学生运用现代数学软件的能力，使教材真正成为基于现代数学软件的、将数学软件融合到具体教学内容中的现代精品教材.

(3) 在内容取舍、材料组织、叙述方式等方面具有较高水准和自身特色.

(4) 数学专业教材要求同步给出重要概念的英文词汇，章末列出中文小结，布置若干道(少量)英文习题，并要求学生用英文解答. 章末列出习题和思考题，并列出可进一步深入阅读的文献. 书末给出中英文对照名词索引.

(5) 公共数学教材具有概括性和简易性，注重强化学生的实验训练和实际动手能力，加强内容的实用性，注重案例分析，提高学生应用数学知识和数学方法解决实际问题的能力.

四、主编职责

丛书组编委员会和出版社确定全套丛书的编写原则、指导思想和编写规范，在这一框架下，每本教材的主编对本书具有明确的责权利：

1. 拟定指导思想

按照丛书的指导思想和特色要求，拟出编写本书的指导思想和编写说明.

2. 明确创新点

教改、课改动态，学科发展前沿，先进理念、知识和方法，如何引入教材；知识和内容创新闪光点及其编写方法；教学实践创新的具体操作；创新与继承的关系把握及其主客体融合.

3. 把握教材质量

质量是图书的生命，保持和发扬科学出版社“三高”、“三严”的传统特色，创出品牌；适用性是教材的生命力所在，应明确读者对象，篇幅要结合大部分学校对课程学时数的要求.

4. 掌握教材编写环节

(1) 把握教材编写人员水平，原则上要求博士、副教授以上，有多年课程教学经历，熟悉课程和学科领域的发展状况，有教材编写经验，有扎实的文字功底.

(2) 充分注意著作权问题，不侵犯他人著作权.

(3) 讨论、拟定教材提纲，并负责编写组的编写分工、协调与组织.

(4) 拟就内容简介、前言、目录、样章，统稿、定稿，确定交稿时间.

(5) 负责出版事宜，敦促编写组成员使用本教材，并优先选用本系列教材.

《21世纪大学数学创新教材》组编委员会

2009年6月

前　　言

数学的应用正向几乎所有的科学领域渗透. 除了自然科学、工程技术、农业科学等领域外,还出乎意料地渗透到语言学、社会学、历史学等许多人文科学和其他领域,运用数学解决实际问题已经显得越来越重要.

“学数学,用数学”一直是我们的教学理念. 1997～2008 年,我们开展了以教学内容和课程体系改革为主体的,以注重增强大学生“用数学”的意识,培养学生“用数学”的能力为目标的教学改革. 改革解决了教学内容与课程体系的设置,建立了完善的数学应用推广机制,每年学习数学建模的人数从 1999 年的 15 人上升到 2008 年的 1400 人. 而且,最近几年我们的国家建模成绩显著提升,初步显现了我们的教学改革成效.

但是我们发现,数学应用能力的培养仍然是大学数学课堂教学中最薄弱的环节之一,缺乏实用有效、便于自学的教材的矛盾日益体现. 为了满足广大师生对数学建模技术的迫切需求,我们编写了这本教材.

本书得到了湖北省教改项目“农林高校数理化基础课实践教学体系的创新与实践”的大力支持,建立提高大学生的实践动手能力与创新能力的教学体系是本项目改革的重点. 数学学习的目标不仅仅是为了锻炼学生的计算能力,更重要的是提高学生运用数学解决实际问题的能力. 要提高这种能力必须要大力推广和普及数学建模方法与数学软件. 本书就是进行这种普及和推广所依赖的重要工具之一.

在本书中,我们依照科学研究、大学生科技创新以及国家建模竞赛所需要的数学建模方法,通过实例与算法程序设计介绍了多元统计、时间序列分析、线性与非线性规划、多目标规划与目标规划、图论、动态规划、排队论、智能优化算法、微分与差分、模糊数学、神经网络、计算机仿真、灰色系统和层次分析法等多种技术. 本书内容详实,通俗易懂,便于自学,其主要特点有:

第一,本书所有程序都在计算机上进行过调试和优化,运行可靠;

第二,本书案例是我们多年来的教学总结,具有代表性;

第三,本书案例丰富、图文并茂、条理清晰;

第四,全书将建模技术与数学实验融为一体,注重数学建模思想介绍,重视数学软件(SAS、MATLAB、LINGO)在实际中的应用.

本书由汪晓银(华中农业大学)、周保平(塔里木大学)担任主编. 全书共 9 章,第 1 章、第 2 章介绍了统计建模与 SAS 编程,由汪晓银、成峰、何丽娟编写;第 3 章介绍了线性规划、非线性规划、多目标规划和目标规划以及 LINGO 编程,由任兴龙、宋双

营编写;第 4 章介绍了图论与 MATLAB 编程,由周保平、任兴龙编写;第 5 章介绍了动态规划与排队论,由李治、齐立美(塔里木大学)编写;第 6 章介绍了智能优化算法,由谭劲英、王邦菊、杨前雨编写;第 7 章介绍了微分方程与差分方程及 MATLAB 编程,由侯志敏、徐思、周林编写;第 8 章介绍了模糊数学方法及 MATLAB 编程,由方红、李雪菲编写;第 9 章介绍了神经网络、计算机仿真、灰色系统和层次分析法,分别由石峰、徐艳玲、汪晓银、胡汉涛(塔里木大学)编写. 汪晓银、周保平审阅全稿,并对全稿进行了排版校对.

为了方便读者更好的学习本书,我们将所有建模技术的软件实现代码放到华中数学建模网 http://www.shumo.cn/大学数学实验栏目,供读者下载.

本书是集体智慧的结晶,是多年来教学的总结. 但限于水平所限,书中难免有错误之处,恳请读者批评指正.

编　者

2009 年 12 月 22 日



目　　录

第1章　多元统计

在工业、农业、医学、气象、环境以及经济、管理等诸多领域中，常常需要同时观测多个指标. 在对多个随机变量的观测数据进行有效地分析和研究时，一种做法是将多个随机变量分开分析，每次分析一个变量；另一种做法是同时进行分析研究. 显然前者做法虽然有时是有效的，但分开处理不仅会丢掉很多信息，往往也不容易取得很好的研究结果. 因此，本章通过定性与定量分析、理论与实例介绍几种常用的多元统计方法.

1.1　多元回归

多元回归分析是研究多个变量之间关系的回归分析方法，按因变量和自变量的数量对应关系可划分为一个因变量对多个自变量的回归分析（简称为“一对多”回归分析）及多个因变量对多个自变量的回归分析（简称为“多对多”回归分析），按回归模型类型可划分为线性回归分析和非线性回归分析. 数学建模中建立变量之间互相影响的模型需要用到这类方法.

1.1.1　多元线性回归

1. 多元线性回归的概念

设自变量 $x_1,x_2,\cdots,x_p$ 的观测值 $x_{i1},x_{i2},\cdots,x_{ip}$ 及因变量 y 对应的观测值 y_i 满足关系式：

$$y_i=\beta_0+\sum_j \beta_j x_{ij}+\varepsilon_i,\quad i=1,2,\cdots,n;j=1,2,\cdots,p$$

其中，$\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_n$ 是相互独立且都服从正态分布 $N(0,\sigma^2)$ 的随机变量，且

$$\boldsymbol{Y}=\begin{bmatrix}y_1\\y_2\\\vdots\\y_n\end{bmatrix},\quad \boldsymbol{X}=\begin{bmatrix}1&x_{11}&x_{21}&\cdots&x_{p1}\\1&x_{12}&x_{22}&\cdots&x_{p2}\\\vdots&\vdots&\vdots&&\vdots\\1&x_{1n}&x_{2n}&\cdots&x_{pn}\end{bmatrix},\quad \boldsymbol{\beta}=\begin{bmatrix}\beta_1\\\beta_2\\\vdots\\\beta_p\end{bmatrix},\quad \boldsymbol{e}=\begin{bmatrix}\varepsilon_1\\\varepsilon_2\\\vdots\\\varepsilon_n\end{bmatrix}$$

根据最小二乘法，得

$$\hat{\boldsymbol{\beta}}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y})$$

2. 回归方程显著性检验

与一元线性回归方程相类似，多元线性回归方程的总离差平方和 SST 也可以分

解为剩余平方和 SSE 与回归平方和 SSR，即

$$SST=SSE+SSR$$

理论上可以证明：当原假设 H_0 为 $\beta_1=\beta_2=\cdots=\beta_p=0$ 且 H_0 成立时，

$$F=\frac{SSR/p}{SSE/(n-p-1)}\sim F(p,n-p-1),\qquad \hat{\sigma}^2=MSE=\frac{SSE}{n-p-1}$$

为 σ^2 的无偏估计量.

因此，给出显著性水平 α，即可进行回归方程的显著性检验.

3. 回归系数的显著性检验

对自变量系数的检验通常使用 t 检验法.

设随机变量 $x_1,x_2,\cdots,x_n$ 对应的系数为 $b_1,b_2,\cdots,b_n$. 各 x_i 都服从正态分布，所以 b_j 也服从正态分布，且

$$E(b_j)=\beta_j,\qquad D(b_j)=\sigma^2 c_{jj},\qquad \frac{b_j-\beta_j}{\sqrt{\sigma^2 c_{jj}}}\sim N(0,1)$$

其中，c_{jj} 是正规方程组的系数矩阵的逆矩阵中第 j 行第 j 列的元素.

可以提出原假设 H_0 为 $\beta_j=0$ 且 H_0 成立时，由 $\frac{SSE}{\sigma^2}$ 服从 $\chi^2(n-p-1)$ 分布推出：

$$F_j=\frac{b_j^2/c_{jj}}{SSE/(n-p-1)}\sim F(1,n-p-1),\qquad t_j=\frac{b_j/\sqrt{c_{jj}}}{\sqrt{SSE/(n-p-1)}}\sim t(1,n-p-1)$$

因此给出显著性水平 α 即可进行回归常数与回归系数的检验，得到各个系数显著的结论. 其中 F 的显著性以 $\Pr>F$ 表示，t 的显著性以 $\Pr>|t|$ 表示，而统计量 $F=t^2$，因此，$\Pr>F$ 与 $\Pr>|t|$ 应是等价的. 通常使用 $\Pr>|t|$ 小于 α 时拒绝原假设 H_0，认为系数不为 0；否者接受原假设 H_0，认为系数为 0，系数没有通过检验.

例 1.1 某品种水稻糙米含镉量 y(mg/kg)与地上部生物量 x_1(10 g/盆)及土壤含镉量 x_2(100 mg/kg)的 8 组观测值见表 1.1. 试建立多元线性回归模型.

表 1.1 某水稻糙米含镉量的观测值

x_1	1.37	11.34	9.67	0.76	17.67	15.91	15.74	5.41
x_2	9.08	1.89	3.06	10.2	0.05	0.73	1.03	6.25
y	4.93	1.86	2.33	5.78	0.06	0.43	0.87	3.86

编写程序如下：

```
/*数据段*/
data ex;  /*表示数据集为ex*/
input x1-x2 y@@;  /*@@表示连续输入数据*/
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/大学数学实验栏目*/
;
```

```
/*程序段*/
proc reg;  /*调用回归模块*/
model y=x1/cli;  /*对 y 关于 x1 做回归,/cli 表示求预测值与预测区间*/
run;
```

运行结果如下：

(1) 回归方程显著性检验.

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	31.46291	31.46291	494.06	<.0001
Error	6	0.38209	0.06368		
Corrected Total	7	31.84500			

Root MSE	0.25235	R-Square	0.9880
Dependent Mean	2.51500	Adj R-Sq	0.9860
Coeff Var	10.03390		

由 Analysis of Variance 表可知，其 F Value=494.06，Pr>F 的值小于 0.0001，远小于 0.05，故拒绝原假设，接受备择假设，认为 y 与 x1，x2 之间具有显著的线性相关关系；由 R-Square 的值为 0.988 可知该方程的拟合度很高，样本观察值有 98.8% 的信息可以用回归方程进行解释，故拟合效果较好，认为 y 与 x1，x2 之间具有显著的线性相关关系.

(2) 参数显著性检验.

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	3.61051	0.95915	3.76	0.0131
x1	1	-0.19828	0.05822	-3.41	0.0191
x2	1	0.20675	0.09769	2.12	0.0879

由 Parameter Estimates 表可知，对自变量 x2，t 检验值为 t=2.12，Pr>|t| 的值等于 0.0879，大于 0.05，因此，接受原假设 $H_0: \beta_2=0$，认为 x2 的系数应为 0，说明 x2 的系数没有通过检验. 为此，需要在程序 model y=x1 x2 中去掉 x2.

再次运行得到如下结果：

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	5.62117	0.16580	33.90	<.0001
x1	1	-0.31911	0.01436	-22.23	<.0001

由 Parameter Estimates 表可知，对常数检验 t 值为 t=33.9，Pr>|t| 的值小于 0.0001，远小于 0.05，说明截距项通过检验，估计值为 5.62117.

对自变量 x1 检验 t 值为 t=-22.23,Pr>|t|的值小于 0.0001,远小于 0.05,说明 x1 的系数通过检验,估计值为-0.31911.

以上结果表明所有变量的系数均通过检验,于是该线性模型即可得到. 然而,许多实际问题中可能还会出现某几个变量的系数并没有通过检验,此时,可以在原程序中的 model y=x1-x2 中去掉没用通过的变量,直到所有的系数均通过检验;或者使用逐步回归方法,让软件自动保留通过检验的变量.

(3) 拟合区间.

Output Statistics

Obs	Dep Var y	Predicted Value	Std Error Mean Predict	95% CL Predict		Residual
1	4.9300	5.1840	0.1496	4.4662	5.9018	-0.2540
2	1.8600	2.0024	0.0922	1.3451	2.6598	-0.1424
3	2.3300	2.5353	0.0892	1.8804	3.1903	-0.2053
4	5.7800	5.3786	0.1567	4.6518	6.1055	0.4014
5	0.0600	-0.0176	0.1447	-0.7294	0.6942	0.0776
6	0.4300	0.5441	0.1258	-0.1459	1.2340	-0.1141
7	0.8700	0.5983	0.1241	-0.0898	1.2864	0.2717
8	3.8600	3.8948	0.1087	3.2224	4.5671	-0.0348

以上为样本的拟合结果,其中 Dep Var y 为因变量的原始值,Predicted Value 为 y 的拟合值,95% CL Predict 为拟合值 95%的拟合区间,Residual 为残差. 例如,第一组原函数值为 4.93,拟合区间为[4.4662,5.9018],残差为-0.254.

综合以上分析可以得到回归方程:

$$y=-0.31911x_1+5.62117$$

1.1.2 多元非线性回归

建立多元非线性回归方程在科学研究中应用广泛,其重要方法是将非线性回归方程转化为线性回归方程. 转化时应首先选择适合的非线性回归形式,并将其线性化,再确定线性化回归方程的系数,最后确定非线性回归方程中未知的系数或参数.

对于实际问题,多元非线性回归模型与一元非线性回归类似,首先应对原始数据作图或者通过经验,观察数据发展情况,选择适当的函数进行拟合.

例 1.2 已知 1978～2006 年全国生产总值 GDP(y)、第一产业生产总值(x_1)、第二产业生产总值(x_2)、工业生产总值(x_3)、第三产业生产总值(x_4),请建立 y 对 x_1～x_4 的回归模型.

通过经济模型经验,选用对数线性模型. 输入命令:

```
data ex;
input y x1-x4@@;
y1=log(y);z1=log(x1);z2=log(x2);z3=log(x3);z4=log(x4);
/*对数据做变化,取自然对数后再做回归分析*/
cards;
```

```
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
;
proc reg;  /*调用回归模块*/
model y1=z1 z2 z3 z4/cli;  /*也可写成 y1=z1-z4/cli;*/
proc corr;var z1-z4;  /*求相关系数矩阵*/
run;
```

运行结果如下：

(1) 回归方程显著性检验.

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	59.29570	19.76523	73270.5	<.0001
Error	25	0.00674	0.00026976		
Corrected Total	28	59.30244			

Root MSE	0.01642	R-Square	0.9999
Dependent Mean	5.09388	Adj R-Sq	0.9999
Coeff Var	0.32243		

由 Analysis of Variance 表可知，其 F Value＝73270.5，Pr＞F 的值小于 0.0001，远小于 0.05，故拒绝原假设，接受备择假设，认为 y1 与 z1，z2，z3，z4 之间具有显著的线性相关关系；由 R-Square 的值为 0.9999 可知该方程的拟合度很高，样本观察值有 99.99%的信息可以用回归方程进行解释，故拟合效果较好.

(2) 参数显著性检验.

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	0.96645	0.06662	14.51	<.0001
z1	1	0.09854	0.04497	2.19	0.0384
z2	1	0.67951	0.07739	8.78	<.0001
z3	1	-0.02654	0.07474	-0.36	0.7256
z4	1	0.24800	0.04929	5.03	<.0001

由 Parameter Estimates 表可知，对自变量 z3 检验 t 值为 t＝－0.36，Pr＞|t|的值等于 0.7256，大于 0.05，因此接受原假设 $H_0:\beta_3=0$ 认为 z3 的系数应为 0，说明 z3 的系数没有通过检验. 为此，需要在程序 model y1＝z1 z2 z3 z4 中去掉 z3.

再次运行得到如下结果：

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	0.96744	0.06539	14.79	<.0001
z1	1	0.09768	0.04412	2.21	0.0362
z2	1	0.65586	0.03873	16.93	<.0001
z4	1	0.24696	0.04833	5.11	<.0001

由 Parameter Estimates 表可知，对常数检验 t 值为 t=14.79，Pr>|t|的值小于 0.0001，远小于 0.05，说明截距项通过检验，估计值为 0.96744.

对自变量 z1 检验 t 值为 t=2.21，Pr>|t|的值等于 0.0362，小于 0.05，说明 z1 的系数通过检验，估计值为 0.09768.

对自变量 z2 检验 t 值分别为 t=16.93，Pr>|t|的值小于 0.0001，远小于 0.05，说明 z2 的系数通过检验，估计值为 0.65586.

对自变量 z4 检验 t 值分别为 t=5.11，Pr>|t|的值小于 0.0001，远小于 0.05，说明 z4 的系数通过检验，估计值为 0.24696.

以上结果表明所有变量的系数均通过检验，于是该线性模型即可得到. 然而，许多实际问题中可能还会出现某几个变量的系数并没有通过检验，此时，可以在原程序中的 model y1=z1 z2 z3 z4 中去掉没用通过的变量，直到所有的系数均通过检验. 或者使用逐步回归方法(在 y1=z1 z2 z3 z4 后加/selection=stepwise 表示逐步回归)，让软件自动保留通过检验的变量.

(3) 拟合区间.

Output Statistics

Obs	Dep Var y1	Predicted Value	Std Error Mean Predict	95% CL Predict		Residual
1	2.8241	2.7816	0.005959	2.7457	2.8176	0.0424
2	3.0626	3.0480	0.008304	3.0101	3.0859	0.0146
3	3.1429	3.1513	0.007156	3.1144	3.1882	-0.008332
4	3.2476	3.2608	0.006719	3.2243	3.2974	-0.0132
5	3.3543	3.3671	0.006031	3.3310	3.4031	-0.0128
6	3.4754	3.4720	0.004876	3.4368	3.5073	0.003344
7	3.5947	3.5910	0.004504	3.5560	3.6261	0.003639
8	3.8084	3.8051	0.004259	3.7702	3.8401	0.003259
9	3.9411	3.9398	0.004937	3.9045	3.9751	0.001286
10	4.1497	4.1746	0.006001	4.1386	4.2106	-0.0249

以上仅列出 10 个样本的拟合结果，其中 Dep Var y1 为因变量的原始值，Predicted Value 为 y 的拟合值，95% CL Predict 为拟合值 95% 的拟合区间，Residual 为残差. 例如，第一组原函数值为 2.8241，拟合区间为[2.7457，2.8176]，残差为 0.0424.

综合以上分析，可以得到线性回归方程：

$$y_1=0.09768z_1+0.65586z_2+0.24696z_4+0.96744$$

又因为 $y_1=\ln(y)$，$z_i=\ln(x_i)$ $(i=1,2,4)$，将方程还原可以得到 y 关于 x_1,x_2,x_4 的函数式：

$$y=2.6312x_1^{0.0977}x_2^{0.656}x_4^{0.247}$$

其中 y 表示全国 GDP 总值，x_1 表示第一产业生产总值，x_2 表示第二产业生产总值，x_4 表示第三产业生产总值.

1.1.3 多元回归方法评价

多元回归分析的优点是可以定量地描述某一现象和某些因素间的线性函数关

系，将各变量的已知值代入回归方程便可求得因变量的估计值(拟合值)，从而可以有效地拟合某种现象的发生和发展. 它既可以用于连续变量，也可用于二分变量(0,1回归). 多元回归的应用有严格的限制. 首先要用方差分析法检验因变量 y 与 m 个自变量之间的线性回归关系有无显著性；其次，如果 y 与 m 个自变量总地来说有线性关系，也并不意味着所有自变量都与因变量有线性关系，还需对每个自变量的偏回归系数进行 t 检验，以剔除在方程中不起作用的自变量，也可以用逐步回归的方法建立回归方程，逐步选取自变量，从而保证引入方程的自变量都是重要的.

多元回归是多元统计分析中的一个重要方法，广泛应用于社会、经济、技术以及众多自然科学领域的研究中. 其中多元线性回归适用于各个指标间数据数量级相差不大，数据间没有多重共线性. 当出现多重共线性的问题，可以使用非线性回归，通过利用一些非线性函数的转化，使转化后的数据可以建立多元线性模型.

1.2 聚类分析

聚类分析又称群分析，它是研究分类问题的一种多元统计方法. 所谓类，通俗地说，就是指相似元素的集合. 那么要将相似元素聚为一类，往往需要将定性和定量的分析结合起来去分类. 通常选取元素的许多共同指标，然后通过分析元素的指标值来分辨元素间的差距，从而达到分类的目的. 由于实际生活中的分类问题很多，聚类分析的研究也就显得十分重要了. 聚类分析可以分为 Q 型分类(样品分类)、R 型分类(指标分类). 本书中介绍的是 Q 型分类.

1.2.1 聚类分析的一般步骤

聚类分析的思想：设集合 G 中有 n 个样本 $X_1, X_2, \cdots, X_n$，首先每个样本独自成为一类，计算类内样本间的距离，将距离最近的两个类聚为一类. 然后再计算新类间的距离，并将最近的两个类聚为一类. 以此循环，直到所有的类聚为一类.

1. 聚类分析前的一般步骤

(1) 确定聚类类型，对样品聚类称 Q 型分类；对变量聚类称 R 型分类.

(2) 数据预处理，如数据变换等.

(3) 为了将样品进行分类，就需要研究样品之间的关系. 通常有两种方法：

① 用相似系数，性质相近的相似系数的绝对值越接近于1，彼此不相关的相似系数的绝对值越接近于0.

② 计算距离，将样品视为 p 维空间的一点，通过计算不同样品的距离，距离接近的点归为一类，距离远的点归为不同类.

(4) 计算距离矩阵或相似性系数矩阵 $\boldsymbol{D}$.

2. 聚类分析的一般步骤(Q型分类)

(1) 每个样本独自成类,$G_i=\{X_i\}$ $(i=1,2,\cdots,n)$.

(2) 由距离矩阵或相似性系数矩阵 $\boldsymbol{D}$,找到当前最小的 D_{ij},并将类 G_i,G_j 合为一类得到一个新类 $G_r=\{G_i,G_j\}$.

(3) 重新计算类间的距离,得到新的矩阵 $\boldsymbol{D}$.

(4) 重复第(2)步直到全部合为一类.

聚类分析方法分为最短距离法(sin)、最长距离法(com)、中间距离法(med)、重心法(cen)、类平均法(ave)、可变类平均法(fle)、可变法和离差平均和法(ward 法) 8 种方法.

例 1.3 从 21 个工厂中抽出同类产品,每个产品测两个指标,见表 1.2,将各厂的质量情况进行分类.

表 1.2 工厂指标观测值

工厂	1	2	3	4	5	6	7	8	9	10	11
指标 1	0	0	2	2	4	4	5	6	6	7	−4
指标 2	6	5	5	3	4	3	1	2	1	0	3
工厂	12	13	14	15	16	17	18	19	20	21	
指标 1	−2	−3	−3	−5	1	0	0	−1	−1	−3	
指标 2	2	2	0	2	1	−1	−2	−1	−3	−5	

输入命令:

```
data ex;input x1 x2 factory$@@;  /*符号$表示定义字符型变量*/
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
;
proc cluster  /*系统聚类*/
data=ex method=ward ccc pseudo outtree=tree;
/*ward法,要使用不同的聚类方法可换对应方法的关键词,如最短距离法 sin.ccc 表示要
   求计算 R²、半偏 R² 和 CCC 统计量.Pseudo 选项要求计算伪 F(PSF)和伪 T² (PST2)统计量*/
id factory;
run;
proc tree data=tree horizontal;  /*水平树*/
id factory;  /*工厂为样本*/
run;
```

程序说明:method=ward 表示聚类使用 ward 离差平方和法. ccc 表示要计算半偏 R^2,R^2 和 CCC 立方聚类标准统计量,这三个统计量和下面的伪 F 和伪 T^2统计量,主要用于检验聚类的效果. 当把数据从 $G+1$ 类合并为 G 类时,半偏 R^2统计量说明

了本次合并信息的损失程度,统计值大表明损失程度大. R^2统计量反映类内离差平方和的大小,统计量大表明类内离差平方和小. CCC 统计量的值大说明聚类的效果好.

Pseudo 说明要计算伪 F 和伪 T^2 统计量. 一般认为,伪 F 统计量出现峰值时所对应的分类是较佳的分类选择. 当把数据从 $G+1$ 类合并为 G 类时,伪 T^2 统计量的值大,说明不应该合并这两类.

运行结果如下:

(1) 聚类过程.

Cluster History

NCL	--Clusters	Joined---	FREQ	SPRSQ	RSQ	ERSQ	CCC	PSF	PST2	Tie
20	f1	f2	2	0.0012	.999	.	.	42.2	.	T
19	f7	f9	2	0.0012	.998	.	.	44.4	.	T
18	f12	f13	2	0.0012	.996	.	.	47.0	.	T
17	f17	f18	2	0.0012	.995	.	.	49.9	.	T
16	f5	f6	2	0.0012	.994	.	.	53.1	.	
15	CL19	f8	3	0.0021	.992	.	.	51.1	1.7	T
14	CL17	f19	3	0.0021	.990	.	.	51.3	1.7	
13	f11	f15	2	0.0025	.987	.	.	51.1	.	
12	f3	f4	2	0.0050	.982	.	.	45.0	.	
11	CL14	f20	4	0.0060	.976	.	.	40.8	3.6	
10	CL15	f10	4	0.0067	.969	.	.	38.8	4.0	
9	CL18	f14	3	0.0071	.962	.	.	38.4	5.7	
8	CL12	CL16	4	0.0106	.952	.	.	36.7	3.4	
7	CL13	CL9	5	0.0141	.938	.	.	35.1	3.9	
6	f16	CL11	5	0.0196	.918	.	.	33.6	6.3	
5	CL20	CL8	6	0.0401	.878	.	.	28.8	8.9	
4	CL6	f21	6	0.0463	.832	.830	0.04	28.0	6.4	
3	CL7	CL4	11	0.1406	.891	.744	-1.2	20.1	12.6	
2	CL5	CL10	10	0.1623	.529	.538	-.13	21.3	19.0	
1	CL2	CL3	21	0.5288	.000	.000	0.00	.	21.3	

Cluster History 表示聚类的具体过程,NCL 表示当前系统存在类的总个数,Clusters Joined 表示当前加入的编号,例如,NCL 等于 20 时,类 1,2 聚为一类,FREQ 表示新类的元素个数. SPRSQ 表示类与类间规格化最短距离,RSQ 表示 R^2 统计量,ERSQ 表示偏 R^2统计量,CCC 统计量值. PSF 为伪 F 统计量,PST2 为伪 T^2 统计量. Tie 表示"节",是指当前类间最小距离不止一个的时候,此时可以任意选择一对最短距离进行聚类,再计算其他类与新类的距离. 从 CCC 统计量的结果可以看出,最大值对应的类数为 4. 从 4 类合并为 3 类时,伪 T^2统计量显著增加,伪 F 统计量显著下降,综合各方面的结果,因此分 4 类最为合适.

(2) 动态聚类图.

动态聚类图(图 1.1)表示每一个步骤聚类的编号,以及类间规格化距离.

综合以上分析,可以得到结果,将工厂分为 4 类,分别为第一类:f1,f2,f3,f4,f5,f6;第二类:f7,f8,f9,f10;第三类:f11,f12,f13,f14,f15;第四类:f16,f17,f18,f19,f20,f21.

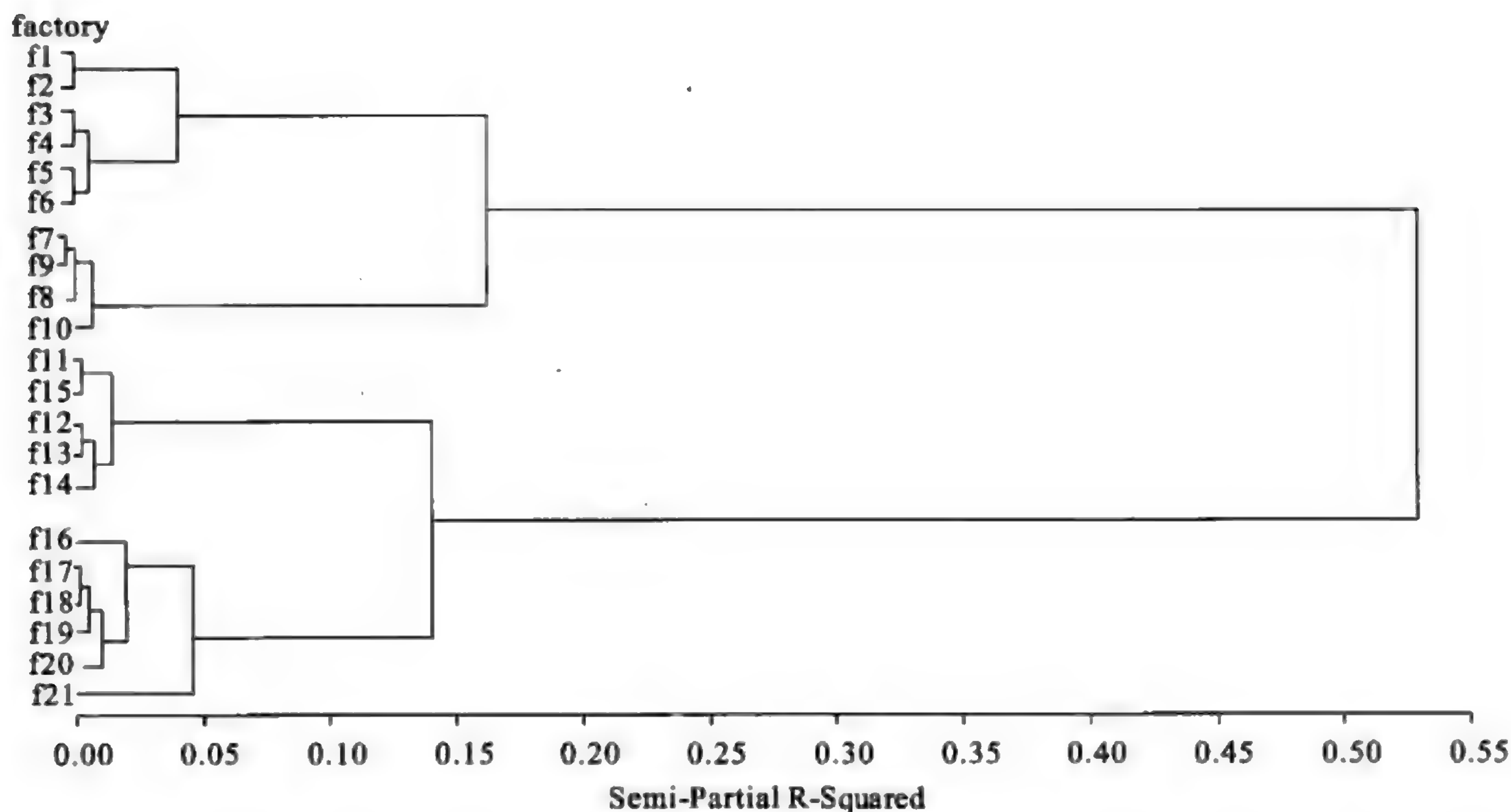


图 1.1 动态聚类图

1.2.2 聚类分析方法的评价

(1) 优点:聚类分析模型的优点就是直观,结论形式简明.

(2) 缺点:在样本量较大时,要获得聚类结论有一定困难.由于相似系数是根据实验对象的反映来建立反映实验对象间内在联系的指标,而实践中有时尽管从实验对象反映所得出的数据中发现他们之间有紧密的关系,但事物之间却无任何内在联系,此时,要根据专业知识来分析聚类分析的结果.

1.3 判别分析

判别分析是基于已知类别的训练样本,对未知类别的样本判类的一种统计方法.与聚类分析不同的是,判别分析是在已知研究对象分成若干类型并已取得各种类型的一批已知样品的观测数据的基础上根据某些准则建立判别式,然后对未知类型的样品进行判别分析.判别分析有 4 类方法:距离判别法、Fisher 判别法、Bayes 判别法和逐步判别法.本书中介绍数学建模中常用的 Bayes 判别法和逐步判别法.

1.3.1 Bayes 判别法的基本思想

Bayes 判别法的基本思想是假设对所研究的对象已有一定的认识,计算新给样品属于各总体的条件概率 $P(l|x)$ $(l=1,2,\cdots,k)$.比较这 k 个概率的大小,然后将新样品判归为来自概率最大的总体.

设有 k 个总体 $G_1, G_2, \cdots, G_k$，它们的先验概率分别为 $q_1, q_2, \cdots, q_k$（它们可以由经验给出也可以估出），各总体的密度函数分别为 $f_1(x), f_2(x), \cdots, f_k(x)$（在离散情况是概率函数），在观测到一个样品 x 的情况下，可用 Bayes 公式计算它来自第 g 总体的后验概率：

$$P(l \mid x) = \frac{q_g f_g(x)}{\sum_{i=1}^{k} q_i f_i(x)}, \quad g = 1,2,\cdots,k$$

且当 $P(h|x) = \max_{1 \leqslant g \leqslant k} P(g|x)$ 时，判 X 来自第 h 总体. 有关理论详见书后参考文献.

1.3.2 Bayes 判别法的一般步骤

综合以上所述，Bayes 判别法的一般步骤如下：

(1) 计算各类中变量的均值 $\overline{x}_j$ 及均值向量 $\boldsymbol{x}_h$ $(h=1,2,\cdots,k)$，各变量的总均值 x_j $(j=1,2,\cdots,p)$ 及均值向量 $\boldsymbol{x}$.

(2) 计算类内协方差矩阵 $\boldsymbol{S}$ 及其逆矩阵 $\boldsymbol{S}^{-1}$.

(3) 计算 Bayes 判别函数中，各个变量的系数及常数项并写出判别函数.

(4) 计算类内协方差矩阵 $\boldsymbol{W}$ 及总各协方差矩阵 $\boldsymbol{T}$ 做多个变量的全体判别效果的检验.

(5) 做各个变量的判别能力的检验.

(6) 判别新样本应属于的类别.

例 1.4 人文发展指数是联合国开发计划署于 1990 年 5 月发表的一份《人类发展报告》中公布的，数据见表 1.3 和表 1.4，试通过已知的数据建立判别函数，误判率是多少？并判断待判的归类.

表 1.3 已知类型的国家指标值

类别	国家	寿命 x_1/年	成人识字率 x_2/%	调整后 GDP x_3/亿美元
1	美国	76	99	5374
1	日本	79.5	99	5359
1	瑞士	78	99	5372
1	阿根廷	72.1	95.9	5242
1	阿联酋	73.8	77.7	5370
1	保加利亚	71.2	93	4250
2	古巴	75.3	94.9	3412
2	巴拉圭	70	91.2	3390
2	格鲁吉亚	72.8	99	2300
2	南非	62.9	80.6	3799

表 1.4　待分类国家指标值

国家	寿命 x_1/年	成人识字率 x_2/%	调整后 GDP x_3/亿美元
中国	68.5	79.3	1950
罗马尼亚	69.9	96.9	2840
希腊	77.6	93.8	5233
哥伦比亚	69.3	90.3	5158

输入命令：

```
data ex;  /*已知类型训练样本*/
input g x1-x3 @@;
cards;
1  76     99    5374  1  79.5   99    5359  1  78    99     5372
1  72.1   95.9  5242  1  73.8   77.7  5370  2  71.2  93     4250
2  75.3   94.9  3412  2  70     91.2  3390  2  72.8  99     2300
2  62.9   80.6  3799
;
data ex1;
input x1-x3 @@;
cards;
68.5      79.3         1950      69.9      96.9         2840
77.6      93.8         5233      69.3      90.3         5159
;
proc discrim  /*判别分析模块*/
data=ex testdata=ex1  /*待判别集合*/
anova manova simple list testout=ex2;  /*判别的结果输出到数据集 ex2*/
class g;
proc print data=ex2;run;
```

程序说明：proc discrim 后的常用选择项有

(1) data＝数据集名，指定输入数据集名，若缺省则指定最新建立的数据集.

(2) testdata＝数据集名，指定待做出判别的数据集名，其中的变量名须与 Data 数据集中的变量名一致.

(3) testout＝数据集名，指定输出数据集，输出 testdata 数据集中所有观测值以及每个观测值的后验概率和判别后的类别.

(4) list，指定打印每个观测值的回代结果.

(5) anova，指定输出各类均值检验的一元统计量.

(6) manova，指定输出各类均值检验的多元统计量.

(7) simple,指定打印总体和组内的简单统计量.

运行结果如下：

(1) 判别函数.

Linear Discriminant Function for g

Variable	1	2
Constant	-323.21568	-236.03823
x1	5.79107	5.14034
x2	0.26498	0.25167
x3	0.03407	0.02533

由以上结果可以得到 Bayes 判别函数为

$$y_1 = -323.21568 + 5.79107x_1 + 0.26498x_2 + 0.03407x_3$$

$$y_2 = -236.03823 + 5.14034x_1 + 0.25167x_2 + 0.02533x_3$$

可以根据计算待判样本的后验概率,同时还可以检验训练样本的类型,得到误判概率,从而得到判别能力.

(2) 误判概率.

Error Count Estimates for g

	1	2	Total
Rate	0.0000	0.0000	0.0000
Priors	0.5000	0.5000	

从上面运行结果得知,Rate 表示对应类的误判概率,Priors 表示对应类的样本总数占样本总数的比例. 两类的误判率均为 0,说明判别能力很强,于是可以利用已经得到的判别函数去判别新样本.

(3) 待判样本分类结果.

Obs	x1	x2	x3	_1	_2	_INTO_
1	68.5	79.3	1950	0.00000	1.00000	2
2	69.9	96.9	2840	0.00000	1.00000	2
3	77.6	93.8	5233	0.99997	0.00003	1
4	69.3	90.3	5159	0.98524	0.01476	1

Obs 表示待判样本的编号,x1,x2,x3 为样本指标值,_1,_2 分别为所属第一类和第二类的概率,_INTO_表示样本所判的类别.

因而得知中国与罗马尼亚归入第二类,希腊与哥伦比亚归入第一类.

1.3.3 逐步判别法

逐步判别法的基本思想与逐步回归相似,每一步选一个判别能力最显著的变化量进入判别函数,并在每选入一个变量之前,对已经选入的变量逐个地检验其判别能

力的显著性，将判别能力已不显著的变量从判别函数中剔出，直到所有可供选择的变量中，既没有变量可以选入，也没有变量可以剔出为止.

逐步判别法的步骤如下：

(1) 计算各总体中各变量的均值和总均值以及似然统计量，规定引入变量和剔除变量的临界值 $F_{进}$ 和 $F_{出}$.

(2) 逐步计算，计算全部变量的判别能力，在已入选变量中考虑剔除可能存在的最不显著变量. 在未选入变量中选出最大判别能力的变量，对变量作 F 检验，通过检验则接受，否则剔除变量. 直到不能剔除又不能增加新变量，逐步计算结束.

(3) 建立判别式，使用第(2)步中选入的变量，用 Bayes 判别法建立判别式.

(4) 对待判样本进行判别分类.

例 1.5 使用逐步判别法求解例 1.4，根据所得结果，对两种方法进行对比.

输入命令：

```
data ex;  /*已知类型训练样本*/
input g x1-x3 @@;
cards;
1  76    99    5374  1  79.5  99    5359  1  78    99    5372
1  72.1  95.9  5242  1  73.8  77.7  5370  2  71.2  93    4250
2  75.3  94.9  3412  2  70    91.2  3390  2  72.8  99    2300
2  62.9  80.6  3799
;
data ex1;
input x1-x3 @@;
cards;
68.5     79.3        1950   69.9     96.9        2840
77.6     93.8        5233   69.3     90.3        5158
;
proc stepdisc data=ex method=stepwise sle=0.3 sls=0.3;
/*选择用逐步判别法,选择后验概率大于 0.3,不注明时系统默认为 0.15*/
class g;var x1-x3;
run;
proc discrim  /*判别分析模块*/
data=ex testdata=ex1  /*待判别集合*/
anova manova simple list testout=ex2;
class g;var x1 x3;  /*选用 x1 和 x3 作为判别指标*/
proc print data=ex2;
run;
```

运行结果如下：

(1) 逐步回归选取变量.

Stepwise Selection Summary

Step	Number In	Entered	Removed	Partial R-Square	F Value	Pr > F	Wilks' Lambda	Pr < Lambda	Average Squared Canonical Correlation	Pr > ASCC
1	1	x3		0.8133	34.85	0.0004	0.18667969	0.0004	0.81332031	0.0004
2	2	x1		0.2701	2.59	0.1516	0.13626590	0.0009	0.86373410	0.0009

由以上结果，通过逐步判别分析选入两个变量 x1，x3 作为判别分析的指标，接下来再以变量 x1，x3 为参考变量作 Bayes 判别.

(2) 得到判别函数.

Linear Discriminant Function for g

Variable	1	2
Constant	-322.01658	-234.95662
x1	6.13904	5.47083
x3	0.03335	0.02465

使用判别分析选出的变量作 Bayes 判别得到判别函数：

$$y_1=-322.01658+6.13904x_1+0x_2+0.03335x_3$$

$$y_2=-234.95662+5.47083x_1+0x_2+0.02465x_3$$

(3) 误判概率.

Error Count Estimates for g

	1	2	Total
Rate	0.0000	0.0000	0.0000
Priors	0.5000	0.5000	

两类的误判率均为 0，说明判别能力很强，于是可以利用已经得到的判别函数去判别新样本.

(4) 待判样本分类结果.

Obs	x1	x2	x3	_1	_2	_INTO_
1	68.5	79.3	1950	0.00000	1.00000	2
2	69.9	96.9	2840	0.00000	1.00000	2
3	77.6	93.8	5233	0.99997	0.00003	1
4	69.3	90.3	5158	0.98420	0.01580	1

结果表明，中国与罗马尼亚归入第二类，希腊与哥伦比亚归入第一类.

由以上两个例子可知，逐步判别法所得到的结果可看出来，尽管这里没有利用变量 x2(成人识字率)，但是最终的判别结果与利用全部变量所得判别结果完全一致，

这说明了三个变量在判别式中所起到的作用不同. 由此可见,在解决现实问题中应结合两种方法使得更加科学地使用已知数据得到更加合理的结论.

1.3.4 判别分析方法的评价

距离判别简单直观,很实用,但是距离判别的方法将总体等同看待,没有考虑到总体会以不同的概率(先验概率)出现,也没有考虑误判之后所造成损失的差异. 一个好的判别方法,既要考虑到各个总体出现的先验概率,又要考虑到错判造成的损失,Bayes 判别就具有这些优点,其判别效果更加理想,应用也更广泛.

1.4 主成分分析

主成分分析是一种通过降维技术将多个变量化为少数几个主成分(即综合变量,通常表示为原始变量的某种线性组合)的统计分析方法. 在数学建模中通常可以用来做数据压缩(降维)、系统评估、回归分析、加权分析等.

1.4.1 主成分分析的概念

主成分分析的基本思想是变量的降维,即将原来众多的具有一定相关性的变量重新组合成一组新的相互无关的综合变量来代替原来的变量. 从而将注意力集中于各次观测中具有最大变异的那些变量,对各次观测中变化不大的变量,可视为常数处理而加以舍弃,从而降低了所考虑问题中变量的维数.

设要进行主成分分析的原指标有 m 个,记为 $x_1,x_2,\cdots,x_m$,现有 n 个样品,相应的观测值为 $x_{ik}\,(i=1,2,\cdots,n)$,而 $k=1,2,\cdots,m$ 作标准化变化后,将 x_k 变化为 x_k^*,即

$$x_k^*=\frac{x_k-\bar{x}_k}{s_k},\quad k=1,2,\cdots,m$$

其中,$\bar{x}_k$ 及 s_k 分别为 x_k 的平均数及标准差,x_k^* 的平均数为 0,标准差为 1.

根据各样本原指标的观测值 x_{ik} 或标准化变化后的观测值 x_{ik}^* 求出系数 b_{kj},建立用标准化变换后的指标 x_k^* 表示综合指标 z_j 的方程 $z_j=\sum\limits_k b_{kj}x_k^*$,也可建立用原指标 x_k 表示综合指标 z_j 的方程 $z_j=\sum\limits_k \tilde{b}_{kj}x_k^*+a_j$.

对于 b_{kj} 的确定依赖于两点要求:

(1) 综合指标间彼此独立或者不相关.

(2) 每个综合指标所反映各个样本的总信息量等于对应特征向量(综合指标的系数)的特征值. 通常要求选取的综合指标的特征值贡献率之和大于 80%.

1.4.2 主成分分析的一般步骤

(1) 由观测数据计算 $\bar{x}_k$ 及 $s_k(k,j=1,2,\cdots,m)$.

(2) 由相关系数矩阵 $\boldsymbol{R}$ 得到特征值 λ_j $(j=1,2,\cdots,m)$ 及各个主成分的方差贡献、贡献率和累计贡献率，并根据累计贡献率确定主成分保留的个数 p.

(3) 写出 m 个基本方程组：

$$\begin{cases} r_{11}x_1^{(j)}+r_{12}x_2^{(j)}+\cdots+r_{1m}x_m^{(j)}=\lambda_j x_1^{(j)} \\ r_{21}x_1^{(j)}+r_{22}x_2^{(j)}+\cdots+r_{2m}x_m^{(j)}=\lambda_j x_2^{(j)} \\ \cdots\cdots \\ r_{m1}x_1^{(j)}+r_{m2}x_2^{(j)}+\cdots+r_{mm}x_m^{(j)}=\lambda_j x_m^{(j)} \end{cases}$$

其中，$j=1,2,\cdots,m$.

利用施密特正交方法，对每一个 λ_j 求它的对应基本方程组的解 $x_1^{(j)},x_2^{(j)},\cdots,x_m^{(j)}$ $(j=1,2,\cdots,m)$，然后令 $b_{kj}=\dfrac{x_k^{(j)}}{\sqrt{\sum_k (x_k^{(j)})^2}}$，从而得到用 $x_1^*,x_2^*,\cdots,x_m^*$ 所表示的主成分 $z_j=\sum_k b_{kj}x_k^*$，或将 $x_k^*=\dfrac{x_k-\bar{x}_k}{s_k}$ 代入后得到用 $x_1,x_2,\cdots,x_m$ 所表示的主成分 $z_j=\sum_k \tilde{b}_{kj}x_k^*+a_j$.

(4) 将 $x_1,x_2,\cdots,x_m$ 的观测值代入主成分的表达式中计算各个主成分的值.

(5) 计算原指标与主成分的相关系数即因子载荷，解释主成分的意义.

例 1.6 为了阐述影响中国蔬菜产量的若干因素，我们收集整理了 1990～2002 年 13 年的数据，请对这些影响因素作主成分分析，并分析结果（选自汪晓银《中国蔬菜生产、消费与贸易研究——一个供需平衡的计量经济分析框架》第 2 章）.

输入命令：

```
data ex;
input x1-x13;
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
;
proc princomp out=prin;  /*主成分分析模块*/
var x1-x13;
run;
proc print data=prin;
var prin1-prin13;
run;
```

从程序结果可以看出，第一、第二、第三主成分累计解释方差的比率已经超过了 94%，所以只需要求 $\lambda_1,\lambda_2,\lambda_3$ 所对应的正交化特征向量 $\boldsymbol{\alpha}_i$ $(i=1,2,3)$.

Eigenvalues of the Correlation Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	10.1634873	8.9130062	0.7818	0.7818
2	1.2504811	0.3718805	0.0962	0.8780
3	0.8786006	0.3466376	0.0676	0.9456
4	0.5319630	0.4530502	0.0409	0.9865
5	0.0789128	0.0282488	0.0061	0.9926
6	0.0506640	0.0153889	0.0039	0.9965
7	0.0352751	0.0297266	0.0027	0.9992
8	0.0055486	0.0024368	0.0004	0.9996
9	0.0031118	0.0014368	0.0002	0.9998
10	0.0016750	0.0015006	0.0001	1.0000
11	0.0001745	0.0000683	0.0000	1.0000
12	0.0001062	0.0001062	0.0000	1.0000
13	0.0000000		0.0000	1.0000

Eigenvectors

	Prin1	Prin2	Prin3	Prin4	Prin5	Prin6	Prin7
x1	0.305923	-.030245	0.033632	-.269456	-.197110	0.154691	0.217163
x2	0.290491	0.229314	-.026216	0.256522	-.566418	0.441434	-.394096
x3	-.218473	0.506586	0.235580	0.496559	-.142694	0.036155	0.590332
x4	0.303059	0.113666	-.078566	0.267547	0.228834	0.143895	-.118972
x5	-.087379	0.773456	0.010433	-.565612	0.191530	0.056548	-.164373
x6	0.313002	0.009982	-.033126	-.045281	0.019068	0.119563	0.142333
x7	0.303651	-.093289	0.018558	-.292412	-.089899	0.157016	0.335570
x8	0.298203	0.133851	-.057994	0.294257	0.522936	-.183988	-.052755
x9	0.299108	0.191451	-.044118	0.068978	-.339437	-.769999	-.197841
x10	0.312148	0.070234	-.030175	-.048472	0.021216	-.060416	0.090627
x11	0.308691	0.040416	-.082388	0.143777	0.347351	0.229396	-.017911
x12	0.309635	-.025691	0.052381	-.143035	-.059903	-.184253	0.426132
x13	0.132852	-.090102	0.958754	-.033753	0.096357	-.008656	-.207248

根据程序结果可知

$$Z_1=\boldsymbol{\alpha}_1\boldsymbol{Y}^{\mathrm{T}},\qquad Z_2=\boldsymbol{\alpha}_2\boldsymbol{Y}^{\mathrm{T}},\qquad Z_3=\boldsymbol{\alpha}_3\boldsymbol{Y}^{\mathrm{T}}$$

其中 $\boldsymbol{Y}=[y_1,y_2,\cdots,y_{13}]$,

$\boldsymbol{\alpha}_1=(0.31,0.29,-0.25,0.30,-0.09,0.31,0.30,0.30,0.30,0.31,0.31,0.31,0.13)$

$\boldsymbol{\alpha}_2=(-0.03,0.23,0.51,0.11,0.77,0.01,-0.09,0.13,0.19,0.07,0.04,-0.03,-0.09)$

$\boldsymbol{\alpha}_3=(0.03,-0.03,0.24,-0.08,0.01,-0.03,0.02,-0.06,-0.04,-0.03,-0.08,0.05,0.96)$.

从程序运行结果可以得到,前面三个主成分 Z_1,Z_2,Z_3 基本上反映了原来所有的信息的 94%. 第一主成分与蔬菜种植面积、每公顷物质费用、蔬菜零售物价指数、市场化程度、城市化水平 1、城市化水平 2、交通、城镇居民可支配收入、农村居民纯收入、农民文化素质等密切相关,表示的是市场经济综合因素,着重反映的是市场经济的成熟程度与国家现代化水平;第二主成分与每公顷劳动投入、成本纯收益率等密切相关,表示的是劳动者动力因素;第三主成分与气候条件密切相关,显然表示的是气候因素.

最后,可以计算三个主成分得分,并将其对应的方差比率作为权重,进行综合评价.

通过对蔬菜产出的影响因素的主成分分析，我们得到影响蔬菜产出水平的主要因素，把原来较多的因素转化成相互独立的彼此不相关的三个主成分来代替原来的诸多因素.

1.4.3 主成分分析方法的评价

(1) 优点：首先，它利用降维技术用少数几个综合变量来代替原始多个变量，这些综合变量集中了原始变量的大部分信息；其次，它通过计算综合主成分函数得分，对客观经济现象进行科学评价. 再次，它在应用上侧重于信息贡献影响力综合评价.

(2) 缺点：当主成分的因子负荷的符号有正有负时，综合评价函数意义就不明确，命名清晰性低，只能涉及一组变量的相关关系，出现两组变量时可以使用后面介绍的典型相关分析.

1.5 因子分析

因子分析又名因素分析法，是在心理学的研究中建立和发展起来的，有人甚至称其为心理学对自然科学的唯一贡献. 20 世纪 70 年代，探索性的因素分析在方法上已经成熟，不仅用于心理学中智力和性格的研究，而且也用于态度、学习等方面的研究，在一些非心理学领域，如化学、地质学、生物学和人文地理学等的研究中也广泛地使用了因素分析方法. 它提供了一种有效的数学模型来解释事物之间的关系.

因子分析也是将具有错综复杂关系的变量综合为数量较少的几个因子，也是多元分析中处理降维的一种统计方法. 数学建模中也常用来做数据压缩、系统评估、加权分析等.

1.5.1 因子分析概念

将主成分分析再往前推进一步就是因子分析. 它试图通过用最少个数的不可观察的变量(公共因子)来说明出现在可观察的变量中的相关模型. 即通过考察每一个变量能否由最少个数的不可观察的公共因子和特殊因子的线性组合来构成：由公共因子生成协方差结构，而特殊因子则说明每一个变量的方差，目的是尽可能合理地解释包含在原始变量之间的相关性，并用以简化变量的维数和结构.

通过变量(或样品)的相关系数矩阵(相似系数矩阵)内部结构的研究，找出控制所有变量(或样品)的少数几个(不可观测的)变量去描述众多原始变量(或样品)之间的相关(或相似)关系.

因子分析(R 型)的数学模型：设 $x_i(i=1,2,\cdots,p)$ p 个变量，如果表示为

$$X_i = a_{i1}F_1 + a_{i2}F_2 + \cdots + a_{im}F_m + \varepsilon_i \quad (m \leqslant p)$$

即 $X = AF + \varepsilon$，$F_1, F_2, \cdots, F_m$ 称为公共因子，是不可观测的变量，$\mathbf{A} = (a_{ij})_{p\times m}$ 称为因子载荷阵，a_{ij} 表示第 i 个变量在第 j 个因子上的载荷(loading)，ε_i 是特殊因子，是不能被前 m 个公共因子包含的部分，并且满足 $\mathrm{Cov}(F,\varepsilon) = 0$，$F$，$\varepsilon$ 不相关.

因子分析解决的三个基本问题：①因子载荷阵 **A** 的估计；②当因子难以得到合理的解释时，对因子载荷阵作正交变换，即因子旋转，目的是对因子的实际意义做出合理的解释；③给出每个变量(或样品)关于 m 个公共因子的得分，通常表示为原始变量的线性组合，即因子得分函数，目的是对公共因子作出估计.

1.5.2 因子分析一般步骤

(1) 类似主成分分析，计算 $\bar{x}_k$ 及 s_k $(k,j=1,2,\cdots,m)$，建立基本方程组.

(2) 用主成分分析法确定因子载荷阵 **A**.

(3) 方差极大正交旋转，对变量系数极值化(尽量趋于 0 或 1).

(4) 得到因子得分函数，计算样本因子得分.

具体操作步骤的推导详见书后参考文献.

例 1.7 已知 12 个地区的 5 个经济指标：人口总数、学校总数、就业人口、服务业总数、房子个数. 依据已知的 5 个经济指标，对 12 个地区的综合经济实力进行分析.

由于指标个数较多，不便于分析排序，因此，考虑先做因子分析找出指标的共同因子，再计算因子得分，通过分析因子得分来评价该地区的经济指标.

输入命令：

```
data ex;
input objects$pop school employ services house@@;
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
;
/*用 proc score 过程输出样品因子得分*/
proc factor data=ex  /*factor 表示调用因子分析模块*/
method=principal rotate=varimax  /*rotate 表示因子旋转*/
percent= 0.8   /*要求累计贡献率大于 0.8*/
score outstat=ex1;  /*计算因子得分*/
var pop school employ services house; run;
proc score data=ex score=ex1 out=ex2;
var pop school employ services house; run;
proc print data=ex1;proc print data=ex2;
run;
```

运行结果如下：

Eigenvalues of the Correlation Matrix: Total = 5 Average = 1

	Eigenvalue	Difference	Proportion	Cumulative
1	2.87331359	1.07665350	0.5747	0.5747
2	1.79666009	1.58182321	0.3593	0.9340
3	0.21483689	0.11490283	0.0430	0.9770
4	0.09993405	0.08467868	0.0200	0.9969
5	0.01525537		0.0031	1.0000

2 factors will be retained by the PROPORTION criterion.

(1) 特征值(因子贡献率).

运行结果显示了 5 个因子对应的特征值、因子贡献率等. 通常确定因子个数时，要求因子累计贡献率大于 80%. 结果表明应选取 2 个因子，记为 F_1, F_2，贡献率分别为 57.47%，35.93%.

(2) 确定因子载荷阵系数，得到初始的特征向量.

Factor Pattern

	Factor1	Factor2
pop	0.58096	0.80642
school	0.76704	-0.54476
employ	0.67243	0.72605
services	0.93239	-0.10431
house	0.79116	-0.55818

由于对应实际问题，公共因子的实际意义不好解释. 因此考虑将指标的系数极值化，即让系数趋于 1 或 0，趋于 1 说明公共因子与该指标密切相关，趋于 0 时说明相关程度很低. 由此，要做因子旋转实现系数的极值化.

因子旋转程序运行结果如下：

The FACTOR Procedure
Rotation Method: Varimax

Orthogonal Transformation Matrix

	1	2
1	0.82069	0.57137
2	-0.57137	0.82069

Rotated Factor Pattern

	Factor1	Factor2
pop	0.01602	0.99377
school	0.94076	-0.00882
employ	0.13702	0.98007
services	0.82481	0.44714
house	0.96823	-0.00605

Orthogonal Transformation Matrix 表示因子旋转阵，Rotated Factor Pattern 为旋转后得到的系数，得到两个因子分别为

$F_1 = 0.016pop + 0.941school + 0.137employ + 0.825services + 0.968house$

$F_2 = 0.994pop - 0.0088school + 0.98employ + 0.447services - 0.006house$

第一公因子 F_1 主要体现房子个数、学校总数和服务业总数;第二公因子 F_2 主要体现就业人口和人口总数.

根据以上得到的因子得分函数,可以计算各个样本两个因子的得分.

(3) 因子得分如下:

The SAS System　　　08:50 Monday, October 5, 2009 148

Obs	objects	pop	school	employ	services	house	Factor1	Factor2
1	A	5700	12.8	2500	270	25000	1.20287	-0.03080
2	B	1000	10.9	600	10	10000	-0.65918	-1.38351
3	C	3400	8.8	1000	10	9000	-1.25937	-0.76740
4	D	3800	13.8	1700	140	25000	1.11491	-0.79697
5	E	4000	12.8	1600	140	25000	0.93710	-0.76320
6	F	8200	8.3	2600	60	12000	-1.22514	0.54857
7	G	1200	11.4	400	10	16000	-0.16819	-1.54932
8	H	9100	11.5	3300	60	14000	-0.44127	0.73444
9	I	9900	12.5	3400	180	18000	0.32032	0.91306
10	J	9600	13.7	3600	390	25000	1.57628	1.02554
11	K	9600	9.6	3300	80	12000	-0.94628	0.96184
12	L	9400	11.4	4000	100	13000	-0.45215	1.10776

以上为 12 个样本的 5 个指标值,以及 2 个公共指标的得分,因子 F1 中 J 地区的得分最高,说明该地区的住房、教育、服务业设施较好. 因子 F2 中 L 地区的得分最高,说明该地区的就业情况较好.

1.5.3 因子分析方法评价

(1) 优点:①它不是对原有变量的取舍,而是根据原始变量的信息进行重新组合,找出影响变量的共同因子,化简数据;②它通过旋转使得因子变量更具有可解释性,命名清晰性高.

(2) 缺点:在计算因子得分时,采用的是最小二乘法,此法有时可能会失效.

1.5.4 因子分析与主成分分析的区别与联系

(1) 因子分析、主成分分析都是重要的降维方法(数据简化技术),因子分析可以视为主成分分析的推广和发展.

(2) 主成分分析不能作为一个模型来描述,它只能作为一般的变量变换,主成分是可观测的原始变量的线性组合;因子分析需要构造因子模型,公共因子是潜在的不可观测的变量,一般不能表示为原始变量的线性组合. 因子分析是用潜在的不可观测的变量和随机影响变量的线性组合来表示原始变量,即通过这样的分解来分析原始变量的协方差结构(相依关系).

1.6 典型相关分析

前面介绍的都是一个随机变量与多个变量之间的线性相关关系. 而现实生活中两组变量间的相关关系的问题很多,例如,投资性变量(如劳动者人数、货物周转量、生产建设投资等)与国民收入变量(如工农业国民收入、运输业国民收入、建筑业国民收入)等. 为此,1936 年由 Hotelling 提出了典型相关分析,揭示了两组多元随机变量之间的关系. 数学建模中也常用典型相关分析研究两组组合相关性的问题.

1.6.1 典型相关分析

其具体做法是:在第一组变量中提出一个典型变量,在第二组量中也提出一个典型变量,并使这一对典型变量具有最大的相关;然后又在每一组变量中提出第二个典型变量,使得在与第一个典型变量不相关的典型变量中,这两个典型变量线性组合之间的相关性最大. 如此下去,直到两组变量间的相关性被提取完毕为止. 可见,典型相关分析是将原来两组变量之间的相关,转化为研究从各组中提出的少数几个典型变量之间的典型相关,从而减少研究变量的个数.

下面通过一个实例来说明典型相关分析的使用方法.

1.6.2 实例分析

例 1.8 蔬菜产出水平主要体现在蔬菜总产量(Y_1)、人均蔬菜占有量(Y_2)、蔬菜总产增长速度(Y_3)三个方面,并称为因变量组(简称"产出组"). 现在的问题是:到底哪个因变量与自变量 X_1,X_2,X_3(简称"影响组")关系更为密切呢? 一般的相关关系很难全面地回答这个问题,解决它最好的办法是典型相关分析法. (选自汪晓银《中国蔬菜生产、消费与贸易研究——一个供需平衡的计量经济分析框架》第 2 章)

输人命令:

```
data ex;
input y1-y3  x1-x3 @@;
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
proc cancorr data=ex all;var y1-y3;   /*cancorr 表示调用相关分析模块*/
with x1-x3;
run;
```

(1) 模型计算与统计检验.

运行结果如下:

将运行结果整理得到分析结果,见表 1.5.

```
                    Canonical Correlation Analysis

                          Adjusted    Approximate        Squared
              Canonical    Canonical      Standard      Canonical
            Correlation  Correlation         Error    Correlation

        1      0.982193     0.977396      0.010189       0.964703
        2      0.810462     0.784388      0.099059       0.656848
        3      0.439231            .      0.232983       0.192924

                                       Test of H0: The canonical correlations in the
          Eigenvalues of Inv(E)*H        current row and all that follow are zero
            = CanRsq/(1-CanRsq)
                                      Likelihood Approximate
  Eigenvalue Difference Proportion Cumulative   Ratio     F Value Num DF Den DF Pr > F

1   27.3309    25.4168     0.9270     0.9270 0.00977553     10.88      9 17.187 <.0001
2    1.9142     1.6751     0.0649     0.9919 0.27694975      3.60      4     16 0.0282
3    0.2390                0.0081     1.0000 0.80707608      2.15      1      9 0.1765
```

表 1.5　蔬菜产出水平与影响因素的三个自变量的典型相关系数及特征值

序号	典型相关系数	标准误差	特征值	方差比率	累计方差比率
1	0.982 193	0.010 189	27.3309	0.9270	0.9270
2	0.810 462	0.099 059	1.9142	0.0649	0.9919
3	0.439 231	0.232 983	0.2390	0.0081	1.0000

从表 1.5 可知，前两个典型相关系数较高，表明相应典型变量之间密切相关. 但要确定典型变量之间的显著性程度，尚需进行相关系数的 F 统计量检验. 其具体做法是：比较 F 计算值与临界值大小，根据比较结果判定典型变量的相关性的显著程度. 其结果见表 1.6.

表 1.6　相关系数检验

序号	F 计算值	自由度	F 检验的显著性概率
1	10.88	9	0.0001
2	3.60	4	0.0282
3	2.15	1	0.1765

从表 1.6 看，只有前两对典型变量通过了统计量检验，表明相应典型变量之间相关关系显著，能够用三个自变量影响变量来解释产出变量.

程序运行得到冗余度分析的结果如下：

```
          Standardized Variance of the WITH Variables Explained by
                     Their Own                              The Opposite
                 Canonical Variables                     Canonical Variables
Canonical
 Variable                   Cumulative   Canonical                  Cumulative
   Number    Proportion     Proportion    R-Square    Proportion    Proportion

        1        0.3335         0.3335      0.9647        0.3217        0.3217
        2        0.3338         0.6672      0.6568        0.2192        0.5409
        3        0.3328         1.0000      0.1929        0.0642        0.6051
```

由以上结果可以得到产出组被典型变量的解释能力(表 1.7).

表 1.7 典型变量的解释能力

序号	产出组方差被影响组解释比例	对产出组解释能力	产出组方差被典型变量解释比例	对影响组解释能力	影响组方差被典型变量解释比例
1	0.9647	0.6481	0.6253	0.3335	0.3217
2	0.6568	0.2054	0.1349	0.3338	0.2192
3	0.1929	0.1465	0.0283	0.3328	0.0642

从表 1.7 可以看出:①前两对典型变量的解释能力均较强;②第一、第二对典型变量具有较高的解释百分比,典型相关系数的平方表明,产出变量中分别有 96.47% 和 65.68%的信息可以由相应的影响变量予以解释;③前两对典型变量的重叠系数较大,产出组的方差被影响组典型变量解释的比例分别为 62.53%,13.49%. 由于第三对典型变量在上述②、③项指标中的数值均较小,且未能通过 F 检验,因此舍弃第三对典型变量,只选定前两对典型变量进行分析.

(2) 典型相关模型.

程序运行可以得到典型相关模型结果如下:

```
                    The CANCORR Procedure

               Canonical Correlation Analysis

     Standardized Canonical Coefficients for the VAR Variables

                      V1              V2              V3

      y1          6.1649         14.7443        -19.9180
      y2         -5.2034        -15.0750         19.9861
      y3          0.0696          0.9105          0.4820

     Standardized Canonical Coefficients for the WITH Variables

                      W1              W2              W3

      x1          0.9953         -0.0132         -0.0962
      x2         -0.0054          0.9591         -0.2831
      x3         -0.0948         -0.2804         -0.9552
```

鉴于原始变量的计量单位不同,不宜直接比较,我们采用标准化的典型系数,给出典型相关模型,见表 1.8.

表 1.8 蔬菜产出水平与影响因素的三个自变量的典型相关模型

序号	典型相关模型
1	$v_1=6.165Y_1-5.203Y_2+0.0696Y_3$ $w_1=0.9953X_1-0.0054X_2-0.0948X_3$
2	$v_2=14.7443Y_1-15.0750Y_2+0.9105Y_3$ $w_2=0.0132X_1+0.9591X_2-0.2804X_3$

(3) 结果分析.

根据典型变量的重要程度及系数大小,从建立的典型相关模型可以看出,三个自变量代表的影响组对蔬菜产出水平的关系可以用两对典型相关变量予以综合描述,具体体现在如下几个方面:

① 自变量 X_1 即市场经济综合因素对中国蔬菜产出水平起根本性作用. 市场经济综合因素与蔬菜总产出的关系体现在第一对典型变量 v_1 和 w_1 中,v_1 是中国蔬菜产出水平各指标的线性组合,其中蔬菜总产出 Y_1 的载荷为 6.165,是各产出水平指标中最大的. w_1 是影响因素指标的线性组合,其中市场经济综合因素(X_1)的载荷为 0.9953,远远超过 w_1 内其他指标的数值. 考虑到第一对典型相关变量的相关系数几乎接近于 1,可以认为,市场经济综合因素对蔬菜总产出水平起根本性作用.

② 自变量 X_2 即劳动力动力因素是决定人均蔬菜占有量的关键因素. 第二对典型变量中. 人均蔬菜占有量 Y_2 在典型变量 v_2 中的载荷为 -15.075,是各产出水平指标中最大的,而自变量 X_2 则在典型变量 w_2 中载荷最大,为 0.9591. 这一对典型相关变量的相关系数非常之高,表明自变量 X_2 对劳动力动力因素起关键作用.

③ 在第二对典型变量中,Y_1 与劳动力因素关系也非常密切. 因为在第二对典型变量中,Y_1 在 v_2 中的载荷 14.7443,与 Y_2 差距并不明显. 由此可以分析得出,用 Y_1 作为产出水平的代表,X_1,X_2,X_3 作为影响变量建立因果拟合模型效果是最好的.

1.6.3 典型相关分析方法评价

(1) 优点:典型相关分析解决了传统多元统计中,只能分析一个变量与多个变量之间关系的问题,实现了两组变量间的分析,可以很好地解决某些组合相关性很高的问题.

(2) 缺点:局限于对两组变量的分析,要求两组变量都是连续变量,其变量都必须服从多元正态分布.

第2章　时间序列分析

所谓时间序列，就是按照时间的顺序把随机事件变化发展的过程记录下来的一列有序数据. 对时间序列进行观察研究，找寻它的发展规律，预测它将来的走势就是时间序列分析. 在日常生产、生活中，时间序列无所不在，时间序列分析有着极为广泛的运用，因此，掌握时间序列分析无疑是十分重要的.

本章主要介绍一元时间序列分析.

2.1　时间序列预处理

在得到一个观察序列后，首先对其平稳性和纯随机性进行检验，这两种检验称为时间序列预处理.

2.1.1　平稳性检验

1. 特征统计量

平稳性是某些事件序列具有的一种统计工具. 要描述这个特征，必须借助以下几个统计工具：

1）概率分布

随机变量族的统计特性完全由它们的联合分布函数或联合密度函数决定. 对于时间序列，定义其概率分布：

$$\{F_{t1,t2,\cdots,tm}(x_1,x_2,\cdots,x_m),\forall m\in(1,2,\cdots,m),\forall t_1,t_2,\cdots,t_m\in T\}$$

2）特征统计量

一个简单的描述时间序列统计特征的方法是研究该序列的低阶矩，特别是均值、方差、自协方差和自相关系数，这些称为特征统计量.

（1）均值　　$\mu_t=E(X_t)=\int_{-\infty}^{+\infty}F_t(x)$

均值反映了时间序列每个时刻的平均水平.

（2）方差　　$D(X_t)=E(X_t-\mu_t)^2=\int_{-\infty}^{+\infty}(x-\mu_t)^2\mathrm{d}F_t(x)$

时间序列的方差反映了序列值围绕着均值做随机波动时的平均波动程度.

（3）自协方差　　$\gamma(t,s)=E(X_t-\mu_t)(X_s-\mu_s)$

（4）自相关系数　　$\rho(t,s)=\dfrac{\gamma(t,s)}{\sqrt{D(X_t)D(X_s)}}$

自协方差和自相关系数表示同一事件在两个不同时期之间的相关程度，就是说度量自己过去的行为对自己现在的影响.

2. 宽平稳时间序列的定义

宽平稳是使用序列的特征统计量来定义的一种平稳性. 它认为序列的统计性质主要由它的低阶矩决定，所以只要保证序列低阶矩（二阶）平稳，就能保证序列的主要性质近似稳定.

满足如下条件的序列$\{x_t\}$称为宽平稳序列：

(i) $E(X_t^2) < +\infty, \forall t \in T$;

(ii) $E(X_t) = \mu$, μ为常数, $\forall t \in T$;

(iii) $\gamma(t,s) = \gamma(k, k+s-t)$, $\forall t, s, k$ 且 $k+s-t \in T$.

3. 平稳时间序列的统计性质

(1) 常数均值 $E(X_t)=\mu, \quad \forall t \in T$

(2) 自协方差函数和自相关函数只依赖于时间的平移长度而与时间的起止点无关：

$$\gamma(k)=\gamma(t,t+k), \quad \forall \text{整数 } k$$

(3) 自相关系数具有规范性、对称性、非负定性、非唯一性.

4. 平稳时间序列的意义

(1) 极大地减少了随机变量的个数，并增加了待估变量的样本容量.

(2) 极大地简化了时序分析的难度，同时也提高了对特征统计量的估计精度.

5. 平稳性的检验

1) 时序图检验

根据平稳时间序列均值、方差为常数的性质，平稳序列的时序图应该显示出该序列始终在一个常数值附近随机波动，而且波动的范围有界、无明显趋势及周期特征.

2) 自相关图检验

平稳序列通常具有短期相关性. 该性质用自相关系数来描述就是随着延迟期数的增加，平稳序列的自相关系数会很快地衰减向零.

例 2.1 检验 1964～1999 年中国纱年产量序列的平稳性.

分析 要了解这若干年中国纱年产量序列的平稳性，只需做时序图检验和自相关图检验即可.

SAS 程序

```
goptions vsize=25cm hsize=25cm;
/*g 表示图,options 表示选择,vsize=25cm 纵向高度为 25*/
data a;  /*a 为数据名*/
input sha@@;  /*@@为自动换行*/
year=intnx('year','1jan1964'd,_n_-1);  /*intnx 间隔取时间变量*/
```

```
format year year4.;  /*年按四位数显示*/
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
run;
proc gplot;  /*画图*/
plot sha*year;  /*纵轴 sha,横轴 year*/
symbol v=circle i=join c=blue;
/*v=circle 表示用小圆圈来表示,i=join 把点连接起来,c=blue 表示蓝色为蓝色*/
proc arima data=a;  /*调用 arima 模块*/
identify var=sha nlag=22;  /*延迟阶数为 22 阶*/
run;
```

从运行结果中,可以得到 1964～1999 年中国纱年产量时序图,如图 2.1 所示.

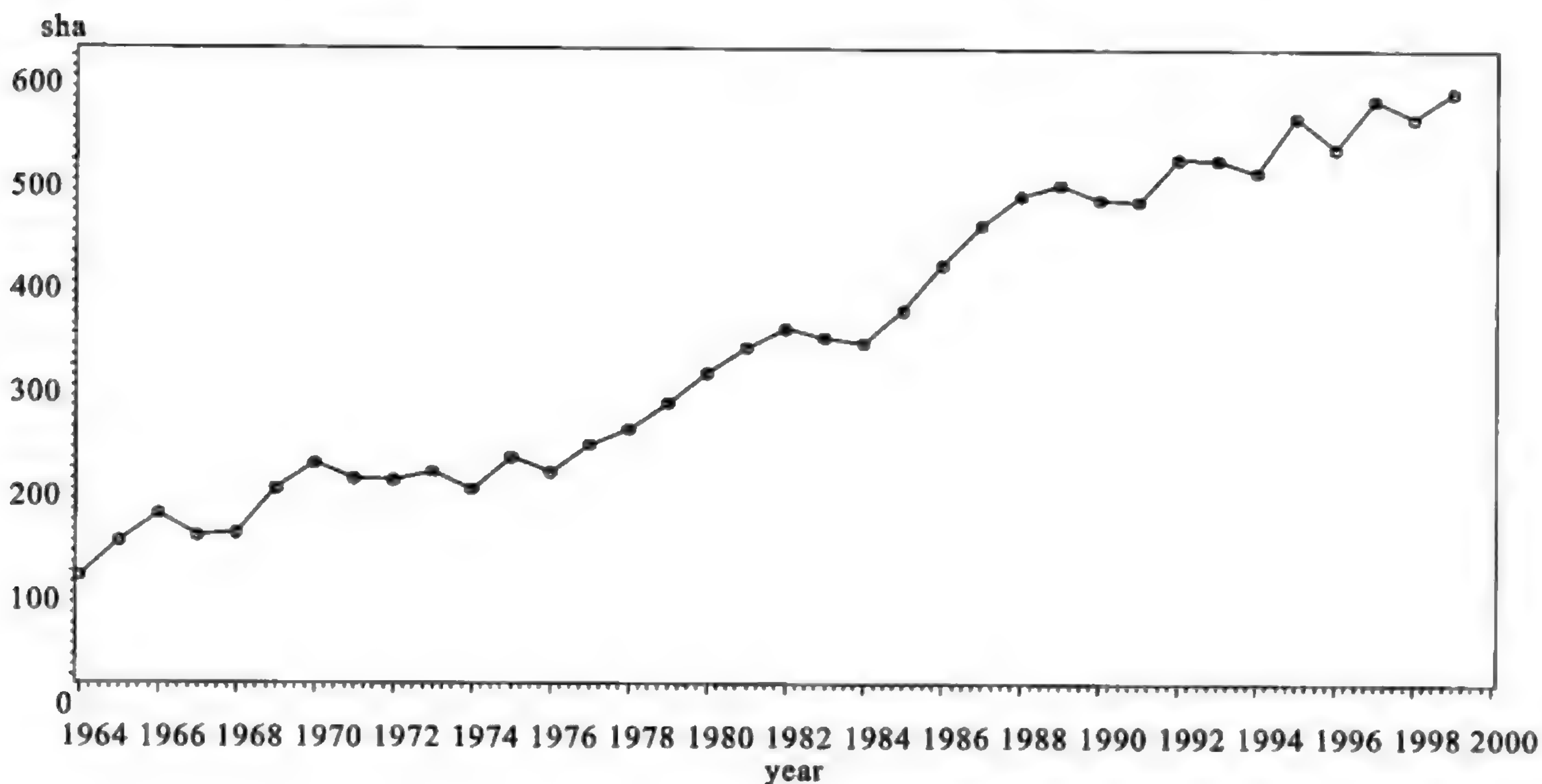
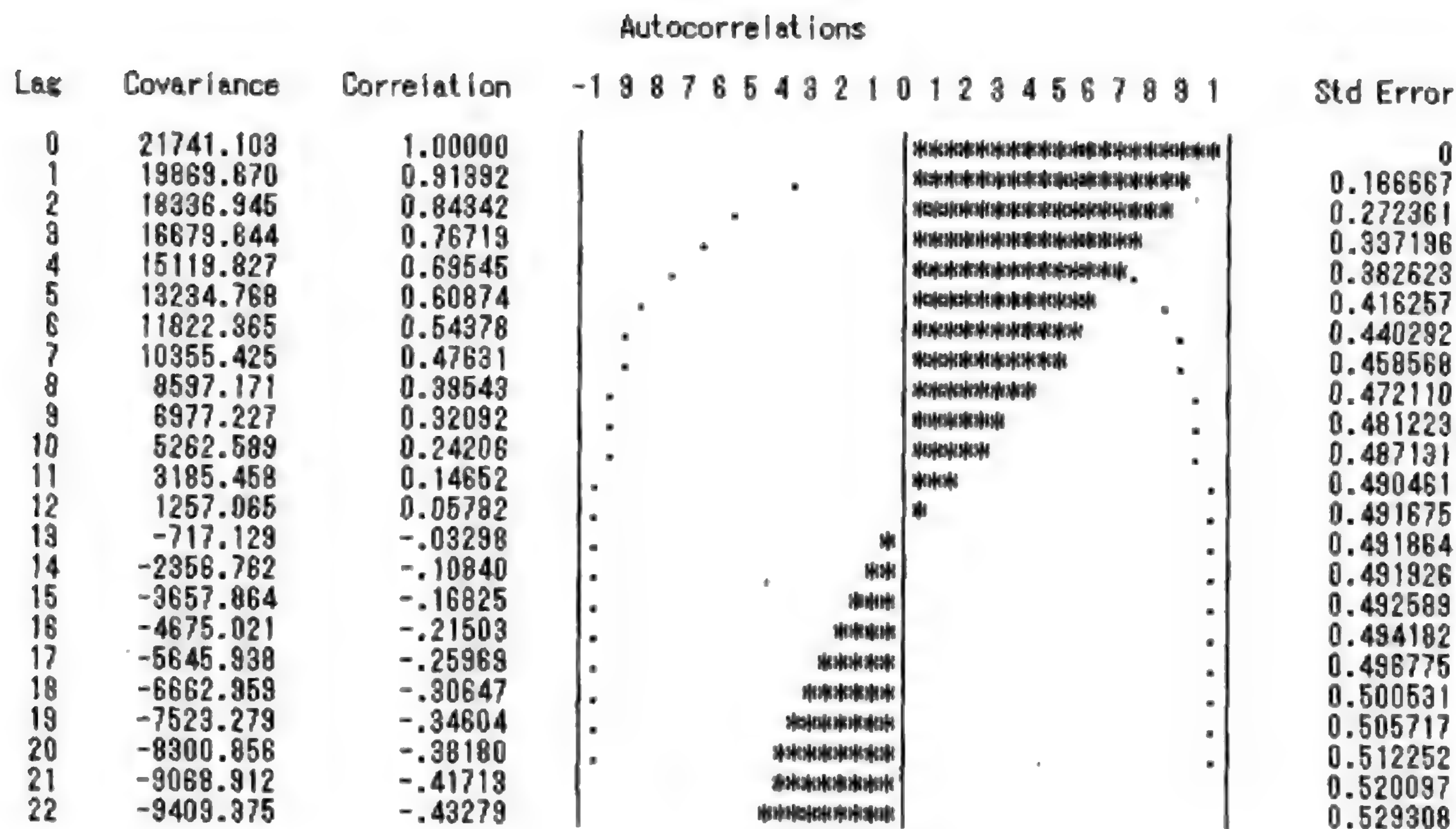


图 2.1　中国纱年产量时序图

从图 2.1 可知,中国纱年产量序列有明显的递增趋势,根据平稳时间序列均值、方差为常数的性质,平稳序列的时序图应该显示出该序列始终在一个常数值附近随机波动,而且波动的范围有界、无明显趋势及周期特征,因此,我们可以得出该序列不是平稳序列.

接着,对中国纱年产量自相关图做分析,从运行结果中,可得出中国纱年产量自相关图如图 2.2 所示. 图 2.2 的横轴表示自相关系数,纵轴表示延迟时期数,用水平方向的垂线表示自相关系数的大小. 从图 2.2 中我们发现中国纱年产量序列的自相关系数从正数缓慢递减到零后又不断在负值范围内增大. 该序列的自相关系数并未较快地衰减向零,因此,该序列不是平稳时间序列.



Autocorrelations

Lag	Covariance	Correlation	Std Error
0	21741.103	1.00000	0
1	19869.870	0.91392	0.166667
2	18336.945	0.84342	0.272361
3	16679.844	0.76719	0.337196
4	15119.827	0.69545	0.382623
5	13234.768	0.60874	0.416257
6	11822.365	0.54378	0.440292
7	10355.425	0.47631	0.458568
8	8597.171	0.39543	0.472110
9	6977.227	0.32092	0.481223
10	5262.589	0.24206	0.487131
11	3185.458	0.14652	0.490461
12	1257.065	0.05782	0.491675
13	-717.129	-.03298	0.491864
14	-2356.762	-.10840	0.491926
15	-3657.864	-.16825	0.492589
16	-4675.021	-.21503	0.494182
17	-5645.938	-.25969	0.496775
18	-6662.959	-.30647	0.500531
19	-7523.279	-.34604	0.505717
20	-8300.856	-.38180	0.512252
21	-9068.912	-.41713	0.520097
22	-9409.375	-.43279	0.529308

图 2.2 中国纱年产量自相关图

综合图 2.1 时序图和图 2.2 自相关图，得出 1964～1999 年中国纱年产量序列的不具有平稳性.

例 2.2 检验 1962 年 1 月～1975 年 12 月平均每头奶牛月产奶量序列的平稳性.

分析 同例 2.1 类似，要了解每头奶牛月产奶量的平稳性，只需做时序图检验和自相关图检验即可.

SAS 程序

```
data a;
input milk@@;
time=intnx('month','1jan1962'd,_n_-1);
format time date.;
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
;
run;
proc gplot;  /*画图*/
plot milk*time;  /*纵轴为 milk,横轴为 time*/
symbol v=square i=join c=red;
/*图形特征.v 表示点的形状,i 表示图形连线的情况,c 代表颜色*/
proc arima data=a;
identify var=milk nlag=22;  /*步长为 22*/
run;
```

从运行结果中，可以得到 1962 年 1 月～1975 年 12 月每头奶牛月产奶量时序图，如图 2.3 所示.

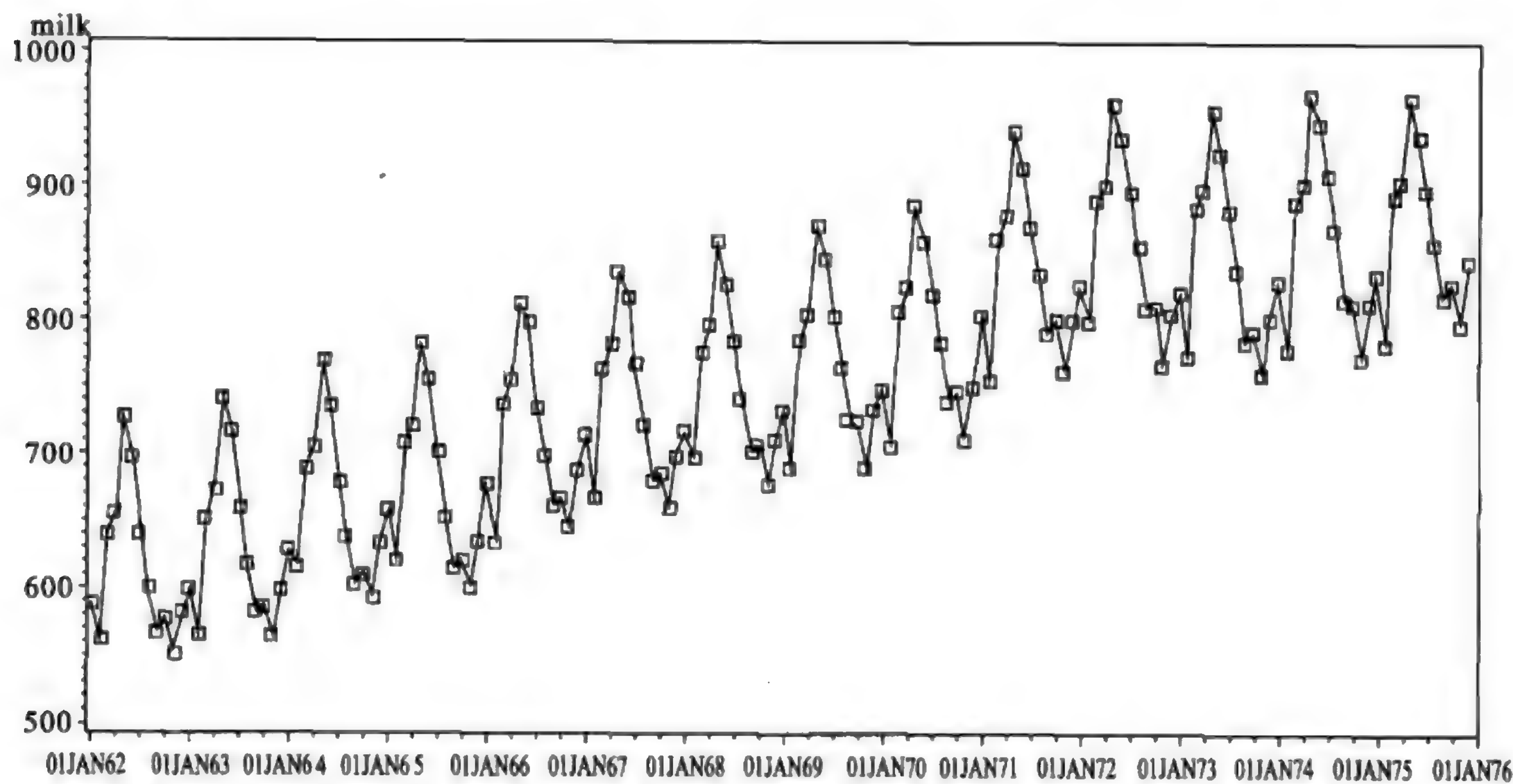
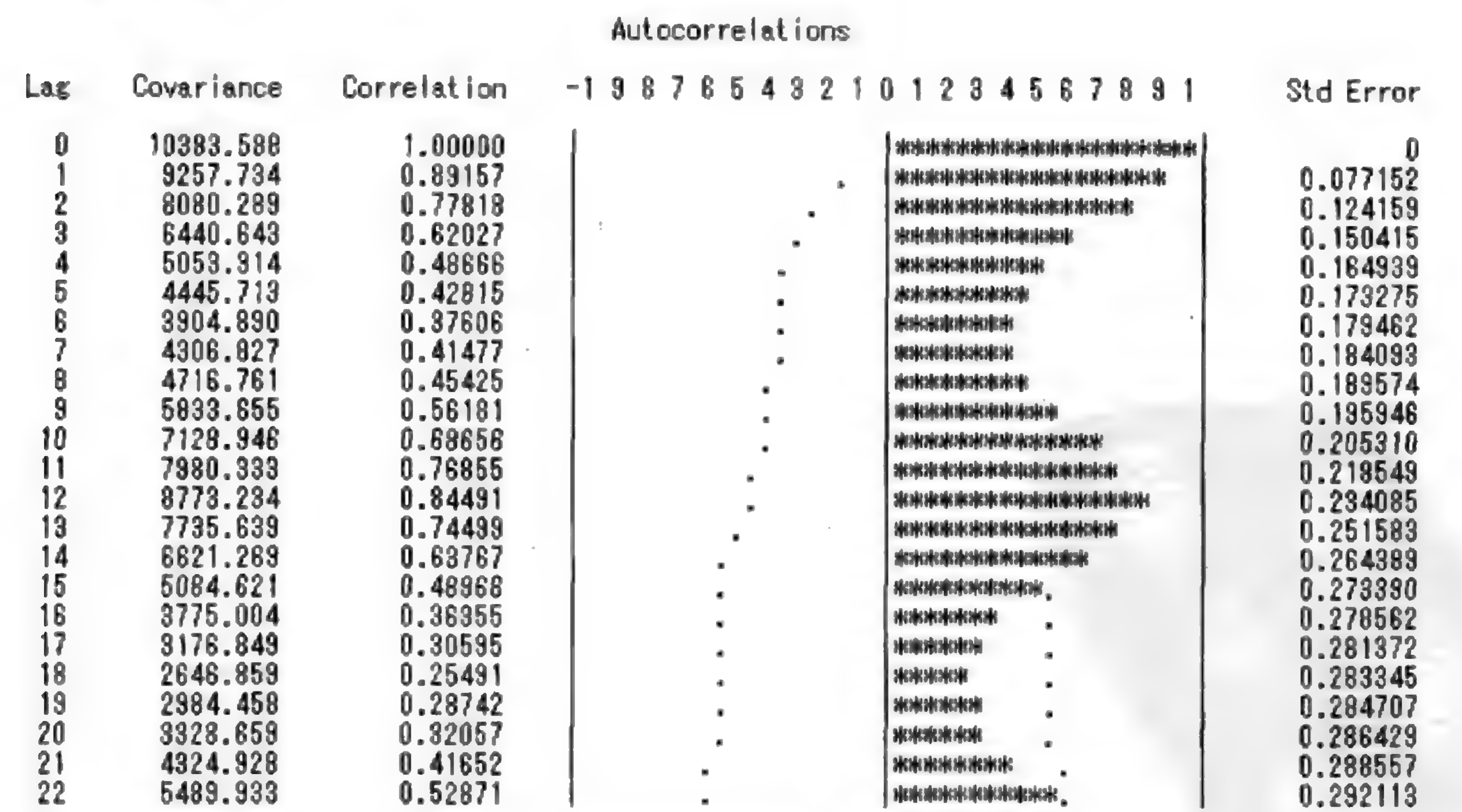


图 2.3　每头奶牛月产奶量时序图

从图 2.3 可知，奶牛月产奶量有递增趋势，由此可见不平稳.

接着，对奶牛月产奶量自相关图做分析，从运行结果中，可得出奶牛月产奶量自相关图，如图 2.4 所示.



Autocorrelations

Lag	Covariance	Correlation	Std Error
0	10383.588	1.00000	0
1	9257.734	0.89157	0.077152
2	8080.289	0.77818	0.124159
3	6440.643	0.62027	0.150415
4	5053.914	0.48666	0.164939
5	4445.713	0.42815	0.173275
6	3904.890	0.37606	0.179462
7	4306.827	0.41477	0.184093
8	4716.761	0.45425	0.189574
9	5833.655	0.56181	0.195946
10	7128.946	0.68656	0.205310
11	7980.333	0.76855	0.219549
12	8773.234	0.84491	0.234085
13	7735.639	0.74499	0.251583
14	6621.269	0.63767	0.264389
15	5084.621	0.48968	0.273390
16	3775.004	0.36355	0.278562
17	3176.849	0.30595	0.281372
18	2646.859	0.25491	0.283345
19	2984.458	0.28742	0.284707
20	3328.659	0.32057	0.286429
21	4324.928	0.41652	0.288557
22	5489.933	0.52871	0.292113

图 2.4　奶牛月产奶量自相关图

从图 2.4 中我们发现奶牛月产奶量的自相关系数表现出周期性,并未衰减为 0. 因此,该序列是不平稳时间序列.

例 2.3 检验 1949～1998 年北京市每年最高气温序列的平稳性.

分析 同例 2.1 类似,要了解这若干年北京市每年最高气温序列的平稳性,只需做时序图检验和自相关图检验即可.

SAS 程序

```
data a;
input  time high;
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
;
run;
proc gplot;
plot high*time;
symbol v=square i=join c=red;
proc arima data=a;
identify var=high nlag=22;
run;
```

从运行结果中,可以得到 1949～1998 年北京市每年最高气温时序图,如图2.5所示.

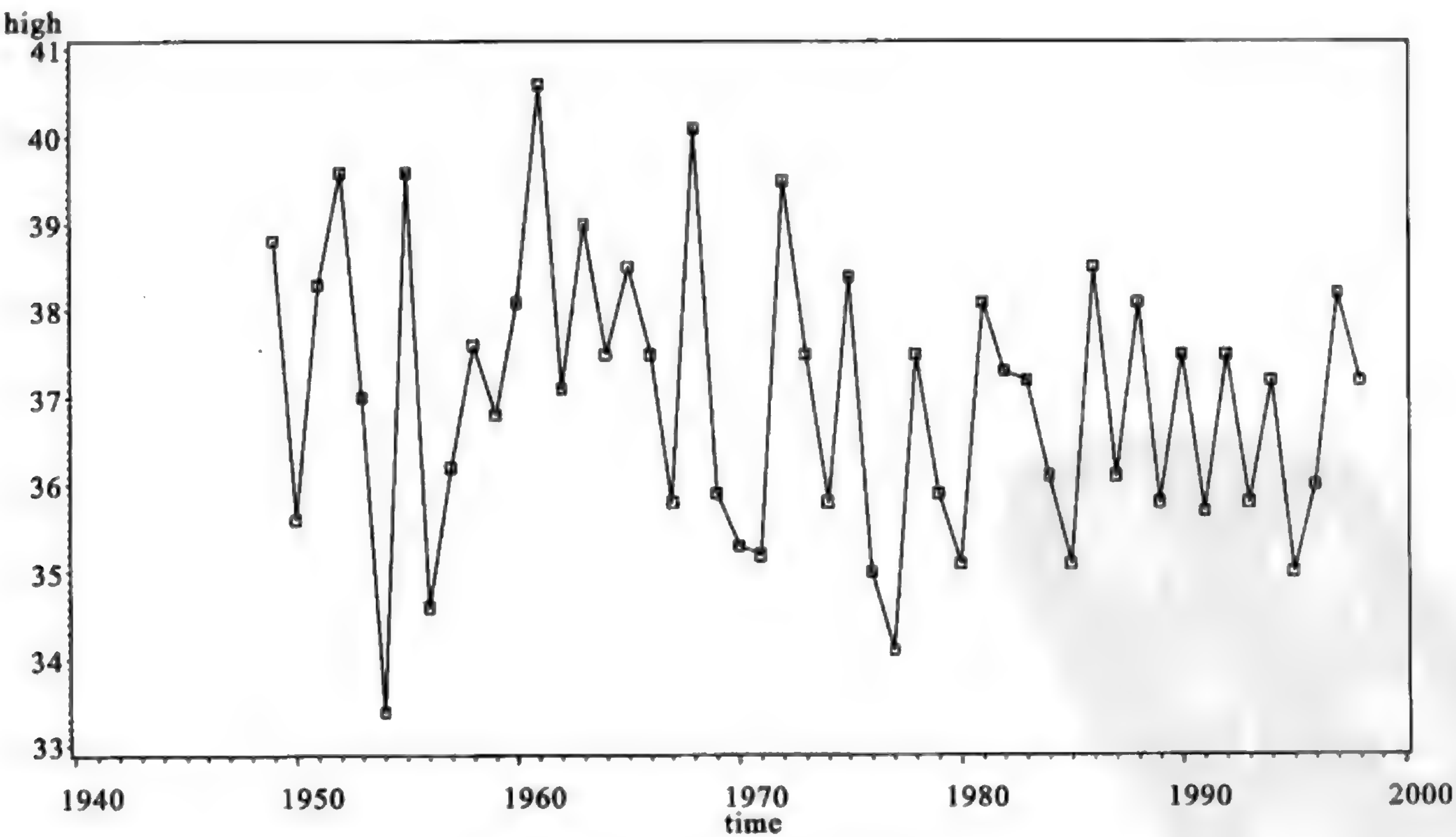
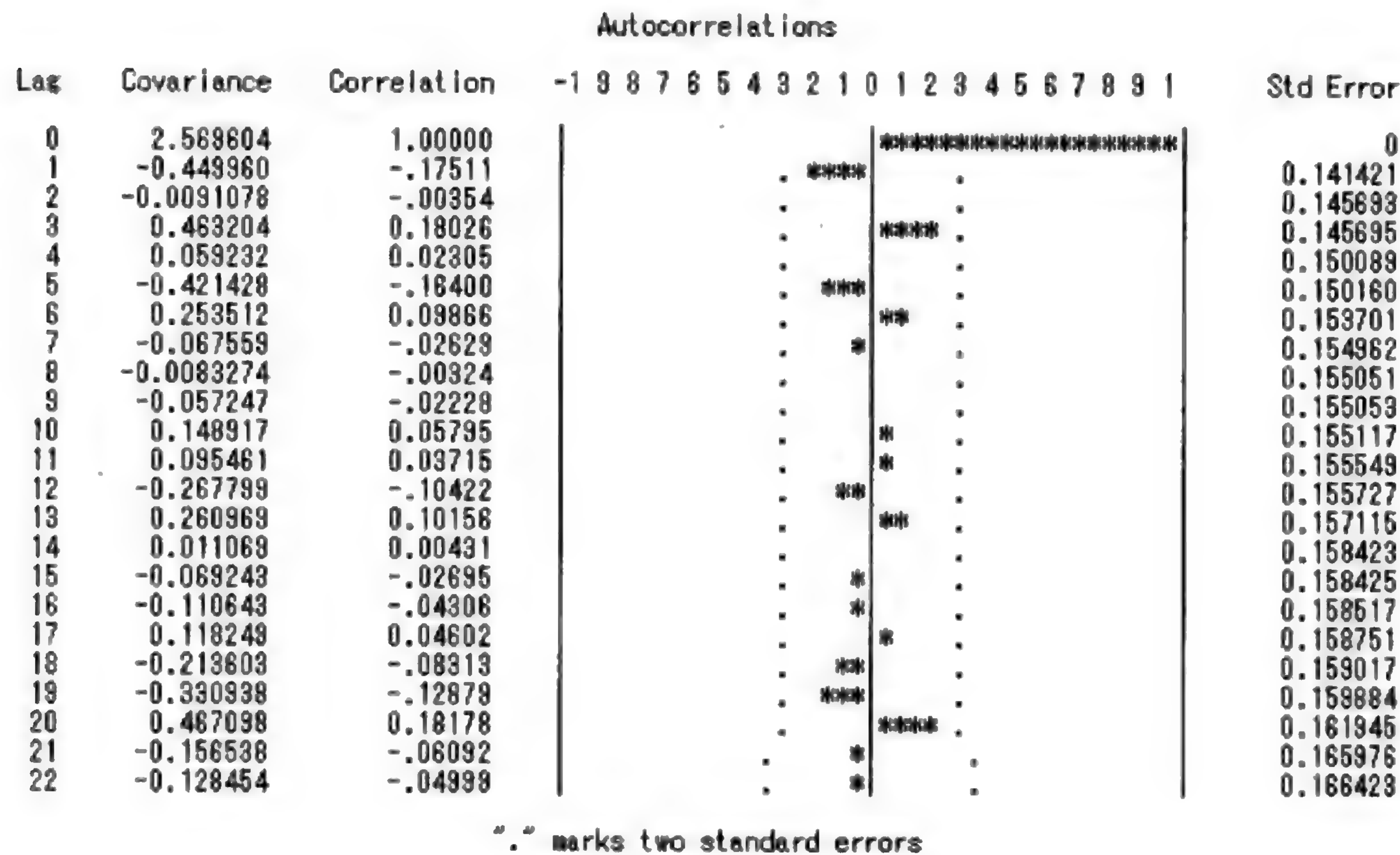


图 2.5 北京市每年最高气温时序图

从图 2.5 可知,北京市每年最高气温始终在一个常数值附近随机波动,而且波动的范围有界、无明显趋势及周期特征,基本可以视为平稳序列. 为了稳妥起见,我们还需要利用自相关图进行进一步辅助识别.

接着,对北京市每年最高气温自相关图做分析,从运行结果中,可得出北京市每年最高气温自相关图,如图 2.6 所示.



Autocorrelations

Lag	Covariance	Correlation	-1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1	Std Error
0	2.569604	1.00000	********************	0
1	-0.449960	-.17511	. **** .	0.141421
2	-0.0091078	-.00354	. .	0.145693
3	0.463204	0.18026	. **** .	0.145695
4	0.059232	0.02305	. .	0.150089
5	-0.421428	-.16400	. *** .	0.150160
6	0.253512	0.09866	. ** .	0.153701
7	-0.067559	-.02629	. * .	0.154962
8	-0.0083274	-.00324	. .	0.155051
9	-0.057247	-.02228	. .	0.155053
10	0.148917	0.05795	. * .	0.155117
11	0.095461	0.03715	. * .	0.155549
12	-0.267799	-.10422	. ** .	0.155727
13	0.260969	0.10156	. ** .	0.157116
14	0.011069	0.00431	. .	0.158423
15	-0.069243	-.02695	. * .	0.158425
16	-0.110643	-.04306	. * .	0.158517
17	0.118249	0.04602	. * .	0.158751
18	-0.213603	-.08313	. ** .	0.159017
19	-0.330938	-.12879	. *** .	0.159884
20	0.467098	0.18178	. **** .	0.161945
21	-0.156538	-.06092	. * .	0.165976
22	-0.128454	-.04998	. * .	0.166423

"." marks two standard errors

图 2.6　北京市每年最高气温自相关图

从图 2.6 中我们发现北京市每年最高气温的自相关系数一直都比较小,自相关系数会很快地衰减向零,且始终控制在 2 倍的标准差范围以内,可以认为该序列自始至终都在零轴附近波动. 因此,该序列是平稳时间序列.

综合图 2.5 时序图和图 2.6 自相关图,得出:检验 1949～1998 年北京市每年最高气温序列具有平稳性.

平稳性检验有两种检验方法:一种是根据时序图和自相关图显示的特征作出图的检验方法;另一种是构造检验统计量进行的假设检验的方法. 图检验法操作简便,运用广泛,但同时也有判断结论带有很强的主观色彩这一缺点. 最好能有统计检验方法加以辅助判断(篇幅有限,本书不予介绍).

在以上介绍的图检验方法,可以根据时序图和自相关图检验任一序列的平稳性,在时序图判断平稳性时应注意该序列是否始终在一个常数值附近随机波动,而且波动的范围有界、无明显趋势及周期特征;在自相关图判断平稳性时应注意是否具有短期相关性,随着延迟期数的增加,序列的自相关系数是否会很快地向零衰减.

2.1.2 纯随机性检验

如果序列彼此之间没有任何相关性，那就意味着该序列是一个完全没有记忆的序列，该序列称为纯随机序列. 纯随机序列没有分析价值，为了确定某平稳序列值不值得继续分析，我们需要对平稳序列进行纯随机性检验.

1. 纯随机序列的定义

纯随机序列也称为白噪声序列，它满足如下两条性质：

(i) $E(X_t)=\mu, \forall t\in T$；

(ii) $\gamma(t,s)=\begin{cases}\sigma^2, & t=s,\\ 0, & t\neq s,\end{cases} \forall t,s\in T.$

2. 纯随机性的性质

(1) 纯随机性：各序列值之间没有任何相关关系，为"没有记忆"的序列，即

$$\gamma(k)=0,\quad \forall k\neq 0$$

(2) 方差齐性：序列中每个变量的方差都相等，即

$$D(X_t)=\gamma(0)=\sigma^2$$

3. 纯随机性检验

纯随机性检验也称白噪声检验，是专门用来检验序列是否为纯随机序列的一种方法.

Barlett 定理 如果一个时间序列是纯随机的，得到一个观察期数为 n 的观察序列，那么该序列的延迟非零期的样本自相关系数将近似服从均值为零、方差为序列观察期数倒数的正态分布：

$$\hat{\rho}\dot{\sim}N\left(0,\frac{1}{n}\right),\quad \forall k\neq 0$$

根据 Barlett 定理，可以构造检验序列的纯随机性.

1) 假设检验

原假设：延迟期数小于或等于 m 期的序列值之间相互独立，即

$$H_0:\rho_1=\rho_2=\cdots=\rho_m=0,\forall m\geqslant 1$$

备择假设：延迟期数小于或等于 m 期的序列值之间有相关性，即

$$H_1:\text{至少存在某个 }\rho_k\neq 0,\forall m\geqslant 1,k\leqslant m$$

2) 检验统计量

(1) Q 统计量 $$Q=n\sum_{k=1}^{m}\hat{\rho}_k^2\sim\chi^2(m)$$

(2) QLB 统计量 $$\text{LB}=n(n+2)\sum_{k=1}^{m}\left(\frac{\hat{\rho}_k^2}{n-k}\right)\sim\chi^2(m)$$

3）判别原则

（1）拒绝原假设. 当检验统计量大于 $\chi^2_{1-\alpha}(m)$ 分位点，或该统计量的 P 值小于 α 时，则可以以 $1-\alpha$ 的置信水平拒绝原假设，认为该序列为非白噪声序列.

（2）接受原假设. 当检验统计量小于 $\chi^2_{1-\alpha}(m)$ 分位点，或该统计量的 P 值大于 α 时，则认为在 $1-\alpha$ 的置信水平下无法拒绝原假设，即不能显著拒绝序列为纯随机序列的假定.

例 2.4 （例 2.3 续）对 1949 ～ 1998 年北京市每年最高气温序列做白噪声检验. （$\alpha=0.05$）

SAS 程序见例 2.3 的 SAS 程序. 选取运行结果中 The ARIMA Procedure 部分.

```
                         The ARIMA Procedure

                   Autocorrelation Check of Residuals

 To       Chi-              Pr >
 Lag     Square     DF     ChiSq    --------------------Autocorrelations--------------------

  6        5.83      5    0.3229    -0.110     0.126     0.131    -0.180     0.170     0.006
 12       10.28     11    0.5050     0.217    -0.009     0.004    -0.092    -0.128    -0.016
 18       11.38     17    0.8361     0.062    -0.006     0.051     0.007    -0.086     0.026
 24       11.85     23    0.9727    -0.042     0.004     0.027    -0.015    -0.050     0.001
```

从运行结果中可得出 LB 统计量检验表，见表 2.1.

表 2.1　LB 统计量检验表

延迟	LB 统计量检验	
	LB 统计量值	P 值
延迟 6 期	5.58	0.4713
延迟 12 期	6.71	0.8760
延迟 18 期	8.36	0.9727

根据表 2.1，由于统计量的 P 值大于 0.05，则认为在 0.05 的显著水平下无法拒绝原假设，即不能显著拒绝序列为纯随机序列的假定. 因而认为北京市最高气温的变动属于纯随机波动，各序列值之间没有任何相关关系，即为“没有记忆”的序列，就是说前后两年北京市最高气温并无大的联系. 这说明我们很难根据历史信息预测未来年份的北京市最高气温，该平稳序列不值得继续分析下去，对该序列分析到此结束.

通过以上的例题分析，我们可以得出以下序列的平稳性与纯随机性检验的基本解题步骤：

（1）绘制该序列时序图.

（2）自相关图检验.

（3）该序列若是平稳序列，进行纯随机性检验. （$\alpha=0.05$）

例 2.5　对 1950～1998 年北京市城乡居民定期储蓄所占比例序列的平稳性与

纯随机性进行检验.

下面通过对例 2.5 的详细分析求解进一步熟悉基本解题步骤.

SAS 程序

```
data a;
input year prop;
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
;
proc gplot;
plot prop*year=1;  /*所画的图记为图 1*/
symbol1 v=diamond i=join c=red;  /*symbol1 表示对图 1 进行操作*/
proc arima data=a;
identify var=prop;  /*变量名为 prop*/
run;
```

步骤一 绘制该序列时序图,如图 2.7 所示.

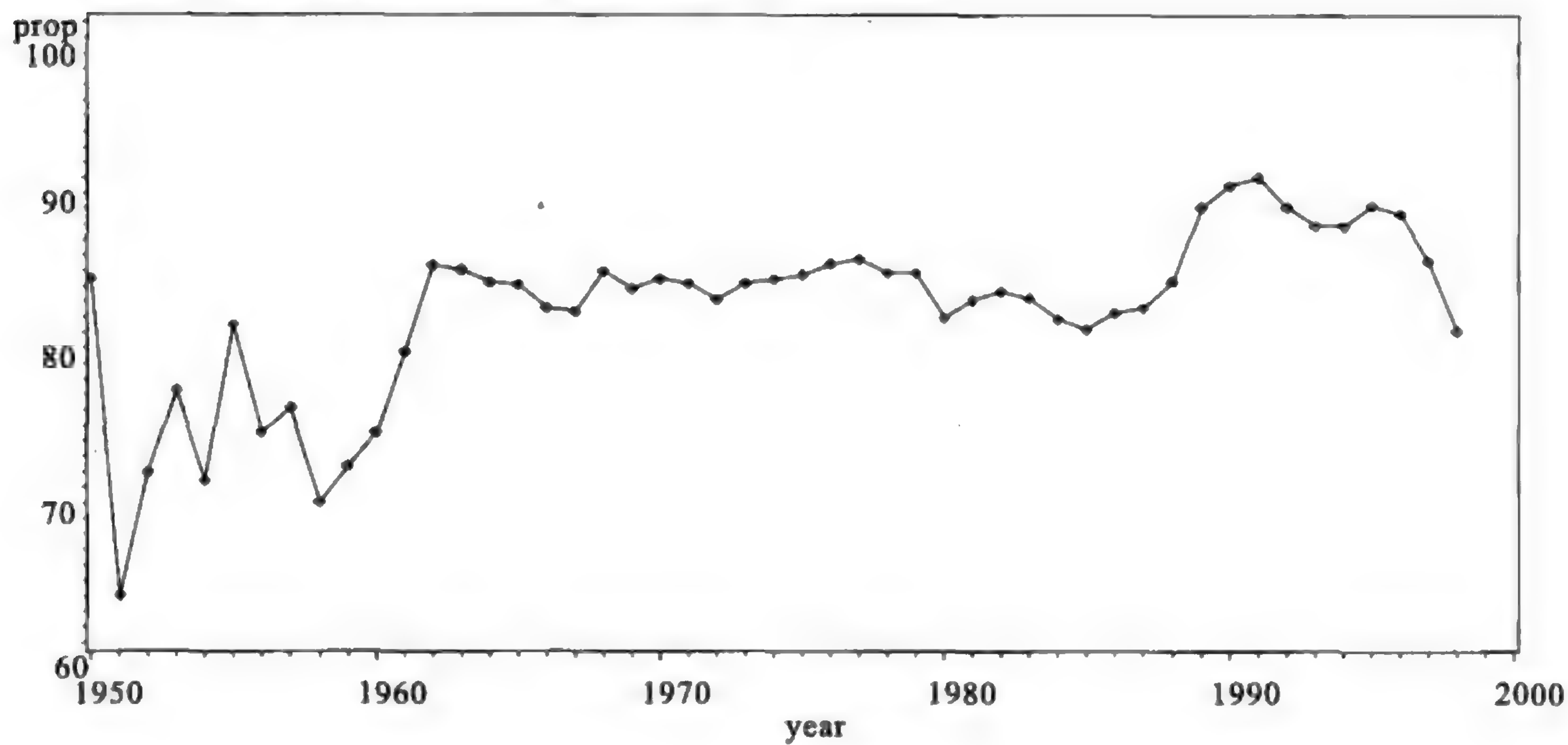


图 2.7 北京市城乡居民定期储蓄所占比例时序图

该序列显示北京市城乡居民定期储蓄所占比例序列波动比较平稳.

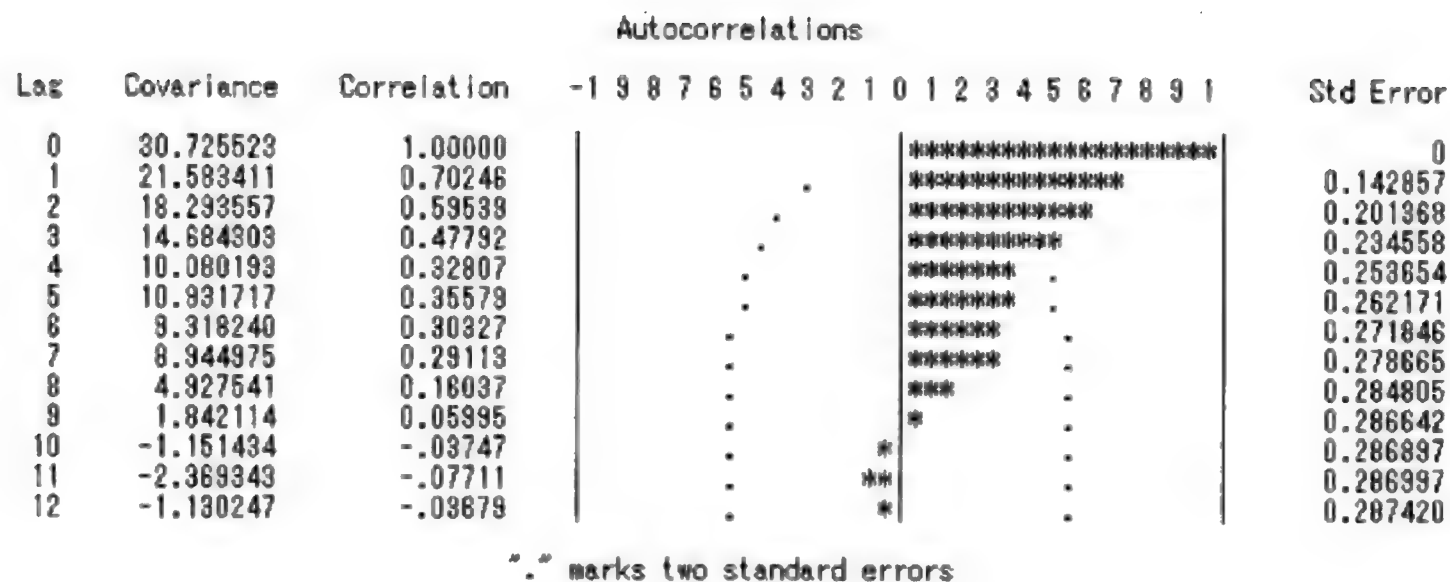
步骤二 绘制该序列自相关图,进一步检验平稳性,如图 2.8 所示.

从图 2.8 可知,样本自相关图延迟 3 阶后,自相关系数都落在 2 倍标准差范围以内,而且自相关系数向零衰减的速度非常快.

结合图 2.7 和图 2.8 可得:北京市城乡居民定期储蓄所占比例为平稳序列.

步骤三 在平稳序列的基础上,对该序列进行纯随机性检验,如图 2.9 所示.

以上结果为白噪声检验结果,其中 To Lag 为延后的阶数,Chi Square 为卡方检验结果,Pr>ChiSq 为统计量的 P 值.



Autocorrelations

Lag	Covariance	Correlation	Std Error
0	30.725523	1.00000	0
1	21.583411	0.70246	0.142857
2	18.293557	0.59539	0.201368
3	14.684303	0.47792	0.234558
4	10.080193	0.32807	0.253654
5	10.931717	0.35579	0.262171
6	9.318240	0.30327	0.271846
7	8.944975	0.29113	0.278665
8	4.927541	0.16037	0.284805
9	1.842114	0.05995	0.286642
10	-1.151434	-.03747	0.286897
11	-2.369343	-.07711	0.286997
12	-1.130247	-.03679	0.287420

"." marks two standard errors

图 2.8　北京市城乡居民定期储蓄所占比例自相关图

Autocorrelation Check for White Noise

To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	75.46	6	<.0001	0.702	0.595	0.478	0.328	0.356	0.303
12	82.87	12	<.0001	0.291	0.160	0.060	-0.037	-0.077	-0.037

图 2.9　白噪声的自相关系数

从图 2.9 中我们可以得出 LB 统计量检验表,见表 2.2.

表 2.2　LB 统计量检验

延迟阶数	LB 统计量检验	
	LB 统计量检验的值	P 值
6	75.46	<0.0001
12	82.57	<0.0001

根据表 2.2,由于统计量的 P 值小于 0.0001,则 P 值远远小于 0.05,则认为在 0.05 的显著水平下拒绝原假设,即拒绝序列为纯随机序列的假定. 因而认为北京市城乡居民定期储蓄所占比例的变动不属于纯随机波动,各序列值之间有相关关系. 这说明我们可以根据历史信息预测未来年份的北京市城乡居民定期储蓄所占比例,该平稳序列属于非白噪声序列.

2.2　平稳时间序列分析

在 2.1 节时间序列预处理中我们提到了如何判别一个序列是否为白噪声序列,当一个序列经过预处理被识别为平稳非白噪声序列,则该序列是一个蕴含着相关信

息的平稳序列. 我们可以通过建立一个模型来拟合该序列,用以提取其中的有用信息.

2.2.1 方法性工具

1. 差分运算

一阶差分:相距一期的两个序列值之间的减法运算称为1阶差分运算.

$$\nabla x_t = x_t - x_{t-1}$$

p 阶差分: 对 $p-1$ 阶差分序列再进行一次1阶差分运算称为 p 阶差分.

$$\nabla^p x_t = \nabla^{p-1} x_t - \nabla^{p-1} x_{t-1}$$

k 步差分: 相距 k 期两个序列值之间的减法运算称为 k 阶差分运算.

$$\nabla_k x_t = x_t - x_{t-k}$$

2. 延迟算子

延迟算子类似于一个时间指针,当前序列值乘以一个延迟算子,就相当于把当前序列值的时间向过去拨了一个时刻. 记 B 为延迟算子,有

$$x_{t-p} = B^p x_t, \quad \forall\, p \geqslant 1$$

延迟算子具有如下的性质:

$B^0 = 1$, $\quad B(cx_t) = cB(x_t) = cx_{t-1}$, $\quad c$ 为任意常数

$B(x_t \pm y_t) = x_{t-1} \pm y_{t-1}$, $\quad B^n x_t = x_{t-n}$

$$\nabla^n x_t = (1-B)^n x_t, \qquad (1-B)^n = \sum_{i=0}^{n} (-1)^i C_n^i B^i, \qquad C_n^i = \frac{n!}{i!(n-i)!}$$

2.2.2 ARMA 模型的性质

ARMA 模型是目前最常用的拟合平稳序列的模型. 它又可以细分为 AR 模型、MA 模型和 ARMA 模型三大类.

1. AR 模型

1) AR 模型的定义

具有如下结构的模型称为 p 阶自回归模型,简记为 AR(p):

$$\begin{cases} x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \varepsilon_t \\ \phi_p \neq 0 \\ E(\varepsilon_t) = 0, \mathrm{Var}(\varepsilon_t) = \sigma_\varepsilon^2, E(\varepsilon_t \varepsilon_s) = 0, s \neq t \\ E(x_s \varepsilon_t) = 0, \forall\, s < t \end{cases}$$

特别地,当 $\phi_0 = 0$ 时,称为中心化 AR(p)模型.

2) 自回归系数多项式

引进延迟算子,中心化 AR(p)模型又可以简记为 $\Phi(B) x_t = \varepsilon_t$,其中

$$\Phi(B)=1-\phi_1 B-\phi_2 B^2-\cdots-\phi_p B^p$$

3）平稳 AR 模型的统计性质

(1) 均值.

如果 AR(p)模型满足平稳性条件，则有

$$E(x_t)=E(\phi_0+\phi_1 x_{t-1}+\cdots+\phi_p x_{t-p}+\varepsilon_t)$$

根据平稳序列均值为常数，且$\{\varepsilon_t\}$为白噪声序列，有

$$E(x_t)=\mu, \qquad E(\varepsilon_t)=0, \quad \forall t\in T$$

推导出
$$\mu=\frac{\phi_0}{1-\phi_1-\cdots-\phi_p}$$

(2) AR 模型自相关系数的性质.

拖尾性：$\rho(k)=\sum_{i=1}^{p}c_i\lambda_i^k$ （$c_1,c_2,\cdots,c_p$ 不能全等于零）呈负指数衰减.

$$\rho(k)=\sum_{i=1}^{p}c_i\lambda_i^k\to 0$$

(3) 偏自相关系数.

定义：对平稳 AR(p)序列，滞后 k 偏自相关系数就是指在给定中间 $k-1$ 个随机变量 $x_{t-1},x_{t-2},\cdots,x_{t-k+1}$ 的条件下，或者说，在剔除了中间 $k-1$ 个随机变量的干扰之后，x_{t-k} 对 x_t 影响的相关度量. 用数学语言描述就是

$$\rho_{x_t,x_{t-k}|x_{t-1},\cdots,x_{t-k+1}}=\frac{E\{[x_t-\hat{E}(x_t)][x_{t-k}-\hat{E}(x_{t-k})]\}}{E[x_{t-k}-\hat{E}(x_{t-k})]^2}$$

偏自相关系数的计算请见相关书籍.

截尾性：AR(p)模型偏自相关系数 p 阶截尾，即

$$\phi_{kk}=0, \quad k>p$$

2. MA 模型

1）MA 模型的定义

具有如下结构的模型称为 q 阶自回归模型，简记为 MA(q)：

$$\begin{cases}x_t=\mu+\varepsilon_t-\theta_1\varepsilon_{t-1}-\theta_2\varepsilon_{t-2}-\cdots-\theta_q\varepsilon_{t-q}\\ \theta_q\neq 0\\ E(\varepsilon_t)=0,\mathrm{Var}(\varepsilon_t)=\sigma_\varepsilon^2,E(\varepsilon_t\varepsilon_s)=0,s\neq t\end{cases}$$

特别地，当 $\mu=0$ 时，称为中心化 MA(q)模型.

2）移动平均系数多项式

引进延迟算子，中心化 MA(q)模型又可以简记为 $x_t=\Theta(B)\varepsilon_t$. 其中

$$\Theta(B)=1-\theta_1 B-\theta_2 B^2-\cdots-\theta_q B^q$$

3）MA 模型的统计性质

常数均值： $E(x_t)=E(\mu+\varepsilon_t-\theta_1\varepsilon_{t-1}-\theta_2\varepsilon_{t-2}-\cdots-\theta_q\varepsilon_{t-q})=\mu$

常数方差：$\mathrm{Var}(x_t)=\mathrm{Var}(\mu+\varepsilon_t-\theta_1\varepsilon_{t-1}-\theta_2\varepsilon_{t-2}-\cdots-\theta_q\varepsilon_{t-q})=(1+\theta_1^2+\cdots+\theta_q^2)\sigma_\varepsilon^2$

4）偏自相关系数拖尾

$$\phi_{kk}=(-\theta_1\varepsilon_{t-1}-\cdots-\theta_q\varepsilon_{t-q})(-\theta_1\varepsilon_{t-k-1}-\cdots-\theta_q\varepsilon_{t-k-q+1})$$

$$\theta_1,\cdots\theta_q \text{ 不恒为零} \Rightarrow \phi_{kk} \text{ 不会在有限阶之后恒为零}$$

3. ARMA 模型

1）ARMA 模型的定义

具有如下结构的模型称为自回归移动平均模型，简记为 ARMA(p,q)：

$$\begin{cases} x_t=\phi_0+\phi_1x_{t-1}+\cdots+\phi_px_{t-p}+\varepsilon_t-\theta_1\varepsilon_{t-1}-\cdots-\theta_q\varepsilon_{t-q} \\ \phi_p\neq 0,\theta_q\neq 0 \\ E(\varepsilon_t)=0,\mathrm{Var}(\varepsilon_t)=\sigma_\varepsilon^2,E(\varepsilon_t\varepsilon_s)=0,s\neq t \\ E(x_s\varepsilon_t)=0,\forall s<t \end{cases}$$

特别地，当 $\phi_0=0$ 时，称为中心化 ARMA(p,q)模型.

2）系数多项式

引进延迟算子，中心化 ARMA(p,q)模型又可以简记为 $\Phi(B)x_t=\Theta(B)\varepsilon_t$. 其中

$$\Phi(B)=1-\phi_1B-\phi_2B^2-\cdots-\phi_pB^p,\qquad \Theta(B)=1-\theta_1B-\theta_2B^2-\cdots-\theta_qB^q$$

3）ARMA(p,q)模型的统计性质

均值：
$$E(x_t)=\frac{\phi_0}{1-\phi_1-\cdots-\phi_p}$$

协方差：
$$\gamma(k)=\sigma_\varepsilon^2\sum_{i=0}^{\infty}G_iG_{i+k}$$

自相关系数：
$$\rho(k)=\frac{\gamma(k)}{\gamma(0)}=\frac{\sum_{j=0}^{\infty}G_jG_{j+k}}{\sum_{j=0}^{\infty}G_j^2}$$

其中 G_j 为 $x_t-\bar{x}$，$\bar{x}$ 是序列的样本均值.

4）ARMA 模型的相关性

综合以上三种模型，得出 ARMA 模型相关性特征表，见表 2.3.

表 2.3　ARMA 模型相关性特征

模型	自相关系数	偏自相关系数
AR(p)	拖尾	p 阶截尾
MA(q)	q 阶截尾	拖尾
ARMA(p,q)	拖尾	拖尾

2.2.3 平稳序列建模

1. 建模步骤

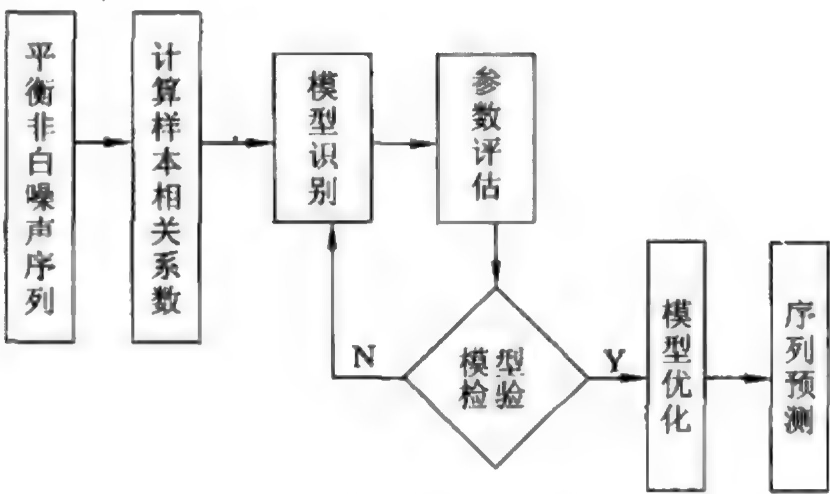


图 2.10 平稳时间序列建模步骤流程图

2. 模型识别

表 2.4 ARMA 模型

自相关系数	偏自相关系数	选择模型
拖尾	p 阶截尾	AR(p)
q 阶截尾	拖尾	MA(q)
拖尾	拖尾	ARMA(p,q)

例 2.6 (例 2.5 续 1)选择合适的模型 ARMA 拟合 1950～1998 年北京市城乡居民定期储蓄比例序列.

根据例 2.5 的分析,得到该序列为非白噪声序列,可以选择合适的模型 ARMA 来拟合该序列.

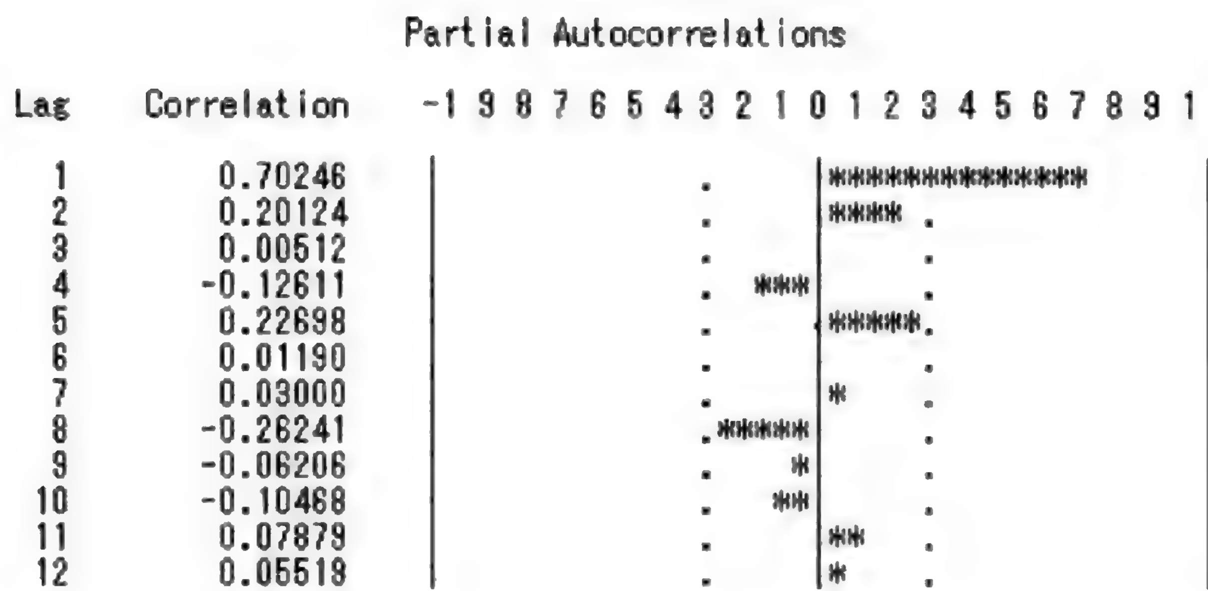
考察该序列的自相关图(见例 2.5 中图 2.8),自相关图中延迟 3 阶后,自相关系数全部衰减到 2 倍标准差范围波动,这表明序列明显的短期相关. 序列由显著的相关系数衰减为小值波动的过程相当缓慢,该自相关系数可视为不截尾.

再考察偏自相关系数图,如图 2.11 所示.

图 2.11 显示除了延迟 1 阶的偏自相关系数显著大于 2 倍标准差之外,其他的偏自相关系数都在 2 倍标准差范围内做小值随机波动,而且由非零相关系数衰减为小值波动过程非常突然,该偏自相关系数可视为 1 阶截尾.

因此,根据表 2.4,我们考虑可以用 AR(1)模型来拟合该序列.

根据自相关图和偏自相关图截尾和拖尾的特性,选择合适的模型 ARMA,MA(q)和 ARMA(p,q)选择过程与 AR(p)类似.



```
                        Partial Autocorrelations

Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  1        0.70246    |                .    |**************      |
  2        0.20124    |                .    |****  .             |
  3        0.00512    |                .    |      .             |
  4       -0.12611    |                .  ***|     .             |
  5        0.22698    |                .    |*****.              |
  6        0.01190    |                .    |      .             |
  7        0.03000    |                .    |*     .             |
  8       -0.26241    |                .*****|     .             |
  9       -0.06206    |                .    *|     .             |
 10       -0.10468    |                .   **|     .             |
 11        0.07879    |                .    |**    .            |
 12        0.05518    |                .    |*     .            |
```

Autocorrelation Check for White Noise

图 2.11　北京市城乡居民定期储蓄比例的样本偏自相关图

3. 参数估计

选择好拟合模型后,下一步就是估计模型中未知参数的值. 对于一个非中心化 ARMA(p,q)模型有

$$\Phi(B)x_t=\Theta(B)\varepsilon_t,\qquad x_t=\mu+\frac{\Theta(B)}{\Phi_p(B)}\varepsilon_t$$

其中

$$\Phi(B)=1-\phi_1 B-\phi_2 B^2-\cdots-\phi_p B^p$$

$$\Theta(B)=1-\theta_1 B-\theta_2 B^2-\cdots-\theta_q B^q$$

该模型共有 $p+q+2$ 个未知参数:$\phi_1,\phi_2,\cdots,\phi_p,\theta_1,\theta_2,\cdots,\theta_q,\mu,\sigma_\varepsilon^2$. 其中 μ 的估计值可以用样本均值估计总体均值得到,对于其他 $p+q+1$ 个未知参数可用矩估计,极大似然估计和最小二乘估计. 我们可以用编写 SAS 程序得到例 2.6 的 AR(1) 模型.

例 2.7　(例 2.5 续 2)确定 1950～1998 年北京市城乡居民定期储蓄比例序列拟合模型.

SAS 程序

```
data a;
input year prop;
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
;
proc gplot;
plot prop*year=1;
symbol1 v=diamond i=join c=red;
proc arima data=a;
```

```
identify var=prop;  /*识别变量*/
estimate p=1 method=ml;
/*p=1,没有 q,说明是 AR 模型,method=ml 表示极大似然估计*/
proc gplot data=out;
run;
```

运行结果如下：

```
Model for variable prop
Estimated Mean    81.55159

Autoregressive Factors
Factor 1:  1 - 0.69141 B**(1)
```

根据以上结果，我们可以写出模型的表达式，Estimated Mean 为 μ 的估计值，即 $\mu=81.55159$，Autoregressive Factors 值为 $\Phi(B)$，$\Phi(B)=1-0.69141B$. 可得 AR(1)模型为

$$x_t=81.55159+0.69141x_{t-1}+\varepsilon_t$$

说明 若选择的模型为 MA(q)模型，则在 SAS 程序中只需修改 estimate p=1 为 estimate q=选择的阶数；若选择的模型为 ARMA(p,q)模型，则在 SAS 程序中只需修改 estimate p=1 为 estimate p=选择的阶数，q=选择的阶数.

4. 模型检验

1）模型的显著性检验

整个模型对信息的提取是否充分.

2）参数的显著性检验

模型结构是否最简. 模型检验的对象为残差序列，目的是为了检验模型的有效性，即对信息的提取是否充分.

(1) 判定原则.

① 一个好的拟合模型应该能够提取观察值序列中几乎所有的样本相关信息，即残差序列应该为白噪声序列.

② 反之，如果残差序列为非白噪声序列，那就意味着残差序列中还残留着相关信息未被提取，这就说明拟合模型不够有效.

(2) 假设条件.

原假设：残差序列为白噪声序列

$$H_0:\rho_1=\rho_2=\cdots=\rho_m=0,\ \forall m\geqslant 1$$

备择假设：残差序列为非白噪声序列

$$H_1:\text{至少存在某个 }\rho_k\neq 0,\ \forall m\geqslant 1,k\leqslant m$$

(3) 检验统计量.

LB统计量：$$\text{LB}=n(n+2)\sum_{k=1}^{m}\left(\frac{\rho_k^2}{n-k}\right)\sim\chi^2(m)$$

例 2.8 (例 2.5 续 3) 确定 1950～1998 年北京市城乡居民定期储蓄比例序列拟合模型的显著性.

在例 2.7 中的 SAS 程序运行结果中选取残差序列白噪声检验.

```
                         Autocorrelation Check of Residuals

 To        Chi-             Pr >
Lag      Square     DF     ChiSq    --------------------Autocorrelations--------------------

  6        5.83      5    0.3229    -0.110     0.126     0.131    -0.180     0.170     0.006
 12       10.28     11    0.5050     0.217    -0.009     0.004    -0.092    -0.128    -0.016
 18       11.38     17    0.8361     0.062    -0.006     0.051     0.007    -0.086     0.026
 24       11.85     23    0.9727    -0.042     0.004     0.027    -0.015    -0.050     0.001
```

图 2.12 残差序列白噪声检验

从图 2.12 我们可以得到残差序列白噪声检验结果，见表 2.5.

表 2.5 残差序列白噪声检验结果

延迟阶数	LB统计量的值	P值
6	5.83	0.3229
12	10.28	0.5050
18	11.38	0.8361
24	11.85	0.9727

从表 2.5 可以分析得到，由于各延迟阶数下 LB 统计量的 P 值都显著大于0.05，可以认为这个拟合模型的残差序列属于白噪声序列，根据模型检验的判别原则，得出该拟合模型显著有效.

(4) 参数显著性检验.

目的：检验每一个未知参数是否显著非零，删除不显著参数使模型结构最精简.

假设条件：$H_0:\beta_j=0$，$H_1:\beta_j\neq0$，$\forall\, 1\leqslant j\leqslant m$

检验统计量：$$T=\sqrt{n-m}\,\frac{\hat{\beta}_j-\beta_j}{\sqrt{a_{jj}Q(\tilde{\beta})}}\sim t(n-m)$$

其中 $Q(\tilde{\beta})$ 为残差平方和方程.

例 2.9 (例 2.5 续 4) 确定 1950～1998 年北京市城乡居民定期储蓄比例序列极大似然估计模型的参数是否显著. ($\alpha=0.05$)

从例 2.7 中的 SAS 程序运行结果中选取参数显著性估计.

```
Maximum Likelihood Estimation

                              Standard                  Approx
Parameter      Estimate          Error    t Value     Pr > |t|     Lag

MU             81.55159        1.76807      46.12       <.0001       0
AR1,1           0.69141        0.10293       6.72       <.0001       1
```

图 2.13　参数显著性检验

从图 2.13 我们可以得到如下表格,见表 2.6.

表 2.6　参数显著性

估计方法	均值 μ 的检验		ϕ_1 的检验		结论
	t 统计量	P 值	t 统计量	P 值	
似然估计	46.12	<0.0001	6.72	<0.0001	两参数检验均显著

从表 2.6 可以清楚地看到对两个参数的检验 t 统计量的 P 值均小于 0.0001,则两参数检验均显著,则每一个未知参数显著非零.

5. 模型优化

1) 问题提出

当一个拟合模型通过了检验,说明在一定的置信水平下,该模型能有效地拟合观察值序列的波动,但这种有效模型并不是唯一的.

2) 优化的目的:选择相对最优模型

当同一个序列可以构造两个拟合模型,两个模型都显著有效,那么到底该选择哪个模型用于统计推断呢?

为了解决这个问题,引进 AIC 准则和 SBC 准则,构造适当的统计量,确定模型相对最优.

AIC 准则　AIC 准则又称最小信息量准则,该准则的指导思想为认为一个拟合模型可以从两个方面去考察:①似然函数值越大越好;②模型未知参数的个数越少越好.

AIC 准则基于其指导思想提出,它是拟合精度和参数个数的加权函数:

$$\text{AIC}=n\ln(\hat{\sigma}_\varepsilon^2)+2(n\text{ 为未知参数个数})$$

使 AIC 函数达到最小的模型被认为是最优模型.

SBC 准则　AIC 准则为选择模型带来很大方便,但是同时也有缺陷. 对于一个序列,在样本容量趋于无穷大时,由 AIC 准则选择的模型不收敛于真实模型,它通常比真实模型所含的未知参数个数要多. 为弥补 AIC 准则不足,Akaike 提出 SBC 统计量. SBC 准则的定义:

$$\text{SBC}=n\ln(\hat{\sigma}_\varepsilon^2)+\ln(n)$$

例 2.10 连续读取 70 个某次化学反应的过程数据，构成一时间序列. 对该序列进行两个模型拟合，并用 AIC 准则和 SBC 准则评判两个拟合模型的相对优劣.

SAS 程序

```
data a;
input yield @@;
time=_n_;
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
;
proc gplot;
plot yield*time;
symbol v=star i=join c=red;
proc arima data=a;
identify var=yield nlag=18;
estimate q=2;  /*在进行模型 AR(1)拟合时,将 q=2 修改为 p=1*/
run;
```

上面已对时序图、自相关图、偏自相关图分析得很详细，故本题省略这些图的给出. 从结果运行可得到自相关图 2 阶截尾，尝试用 MA(2)模型拟合. 使用最小二乘法，确定 MA(2)模型的口径：

$$x_t = 51.17301 + (1 + 0.32286B + 0.31009B^2)\varepsilon_t$$

根据偏自相关系数 1 阶截尾，尝试用 AR(1)模型拟合. 使用最小二乘法，确定 AR(1)模型的口径：

$$x_t = 51.26169 + \frac{\varepsilon_t}{1 + 0.42481B}$$

MA(2)模型和 AR(1)模型残差均通过白噪声检验，模型显著有效且各参数显著性检验均通过. 出现上述情况时，我们可以通过用 AIC 准则和 SBC 准则评判两个拟合模型的相对优劣.

选取用 MA(2)拟合时，可得到图 2.14.

```
Constant Estimate        51.17301
Variance Estimate        119.5653
Std Error Estimate       10.93459
AIC                      536.4556
SBC                      543.2011
Number of Residuals            70
* AIC and SBC do not include log determinant.
```

图 2.14　MA(2)拟合的 AIC 和 SBC 值

从图 2.14 可知，选取用 MA(2)拟合时，AIC 值为 536.4556，SBC 值为 543.2011.

选取用 AR(1)拟合时，可得到图 2.15.

```
Constant Estimate        73.03829
Variance Estimate        120.0735
Std Error Estimate       10.95781
AIC                      535.7896
SBC                      540.2866
Number of Residuals            70
* AIC and SBC do not include log determinant.
```

图 2.15　AR(1)拟合的 AIC 和 SBC 值

从图 2.15 可知，选取用 AR(1)拟合时，AIC 值为 535.7896，SBC 值为 540.2866.
结合图 2.14 和图 2.15，得出表 2.7.

表 2.7　AIC 值和 SBC 值

模型	AIC	SBC
MA(2)	536.4556	543.2011
AR(1)	535.7896	540.2866

根据 AIC 准则，AIC 函数达到最小的模型被认为是最优模型，从这方面来说，AR(1)优于 MA(2)；根据 SBC 准则，SBC 函数达到最小的模型被认为是最优模型，从这方面来说，AR(1)优于 MA(2). 综合这两种准则，本题中 AR(1)模型是相对最优模型.

为了尽量避免因个人不足导致的模型识别问题，SAS 系统还提供了相对最优模型识别. 只要我们在 identify 命令中增加一个可选择命令 minic，就可以获得一定范围内最优模型定阶. 即增加语句：

```
identify var=x nlag=8 minic p=(0:5) q=(0:5)
```

再根据 BIC 最小信息值选择模型.

例 2.11　现有 201 个连续的生产记录，如果已经知道该序列平稳且非白噪声，选择适当模型拟合该序列的发展并写出拟合模型，最后预测该序列后 5 年的 95% 预测的置信区间.

分析　在拟合模型这一块可以用 SAS 系统提供的相对最优模型识别，预测 5 年时可增加语句：

```
forecast id=year lead=5 out=out
```

SAS 程序

```
data a;
input factory@@;
time=_n_;
cards;
/*数据省略，见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
;
```

```
proc gplot;
plot factory*time;
symbol v=diamond i=join c=blue;
proc arima data=a;
identify var=factory nlag=18 minic p=(0:5) q=(0:5);/*模型定阶*/
run;
```

从运行结果可得图 2.16.

Minimum Information Criterion

Lags	MA 0	MA 1	MA 2	MA 3	MA 4	MA 5
AR 0	2.077317	1.960692	1.962695	1.98741	1.97816	1.997981
AR 1	2.00286	1.984271	1.974117	1.992579	1.981907	2.008167
AR 2	1.973647	1.992389	1.998201	2.018435	2.005698	2.031703
AR 3	1.96658	1.990919	2.007238	2.030402	2.032079	2.057841
AR 4	1.970913	1.99316	2.011654	2.03623	2.057557	2.077401
AR 5	1.981602	2.003231	2.029475	2.042908	2.064946	2.090367

Error series model: AR(7)
Minimum Table Value: BIC(0,1) = 1.960692

图 2.16　BIC 最小信息值

从图 2.16 中可知,BIC 最小信息值为 1.960692,根据 BIC 最小信息准则,选择 MA(1)模型是相对最优的. 在选择好模型之后,为得到拟合模型表达式及预测该序列后 5 年的 95%预测的置信区间,继续编写程序.

SAS 程序

```
data a;
input factory @@;
year=_n_;
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
;
proc gplot;plot factory*year;symbol v=diamond i=join c=blue;
proc arima data=a;identify var=factory;estimate q=1 method=ml;forecast
     id=year lead=5 out=out;
proc gplot data=out;plot factory*year=2 forecast*year=3 l95* year=4
     u95*year=4/overlay;
symbol2 v=star i=none c=black;symbol3 v=none i=join c=red;symbol4 v=
     none i=join c=green;
run;
```

运行结果如下:

```
The ARIMA Procedure
Model for variable factory
Estimated Mean      84.1297

Moving Average Factors
Factor 1:  1 - 0.47763 B**(1)
```

由运行结果可知，模型为

$$x_t=84.1297+(1-0.47763B)\varepsilon_t$$

```
Forecasts for variable factory
Obs     Forecast    Std Error     95% Confidence Limits

202     85.6807      2.6898       80.4089      90.9525
203     84.1297      2.9808       78.2874      89.9720
204     84.1297      2.9808       78.2874      89.9720
205     84.1297      2.9808       78.2874      89.9720
206     84.1297      2.9808       78.2874      89.9720
```

图 2.17 预测该序列后 5 年的 95%预测的置信区间值

从图 2.17 可知，预测该序列后 5 年的 95%预测的置信区间可整理成如下表格，见表 2.8.

表 2.8 该序列预测 5 年后的值

预测年数	预测值	95%置信区间下限	95%置信区间下限
1	85.6807	80.4089	90.9525
2	84.1297	78.2874	89.9720
3	84.1297	78.2874	89.9720
4	84.1297	78.2874	89.9720
5	84.1297	78.2874	89.9720

2.3 非平稳序列序列分析

前面主要介绍了对平稳时间序列的分析方法.实际上，在自然界中绝大部分序列都是非平稳的，因而对非平稳序列的随机分析学习是十分重要的.

2.3.1 差分运算

1. 差分运算的实质

差分方法是一种非常简便、有效的确定性信息提取方法. Cramer 分解定理在理

论上保证了适当阶数的差分一定可以充分提取确定性信息.差分运算的实质是使用自回归的方式提取确定性信息.

$$\nabla^d x_t = (1-B)^d x_t = \sum_{i=0}^{d} (-1)^i C_d^i x_{t-i}$$

2. 差分方式的选择

在实践操作过程中,我们会根据序列不同的特点选择合适的差分方式,常见情况有以下三种:

(1) 序列蕴含着显著的线性趋势,一阶差分就可以实现趋势平稳(上升或者下降).

例 2.12 1964~1999 年中国纱年产量序列蕴含着一个近似线性的递增趋势.对该序列进行一阶差分运算.

分析 中国纱年产量序列蕴含着显著的线性趋势,一阶差分就可以实现趋势平稳.该序列平稳之后才能对其做其他一系列处理.

SAS 程序

```
data a;input year sha;dif=dif(sha);  /*一阶差分*/
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
;
proc gplot;plot sha*year;plot dif*year;symbol v=star c=blue i=join;
run;
```

原序列时序图如图 2.18 所示.

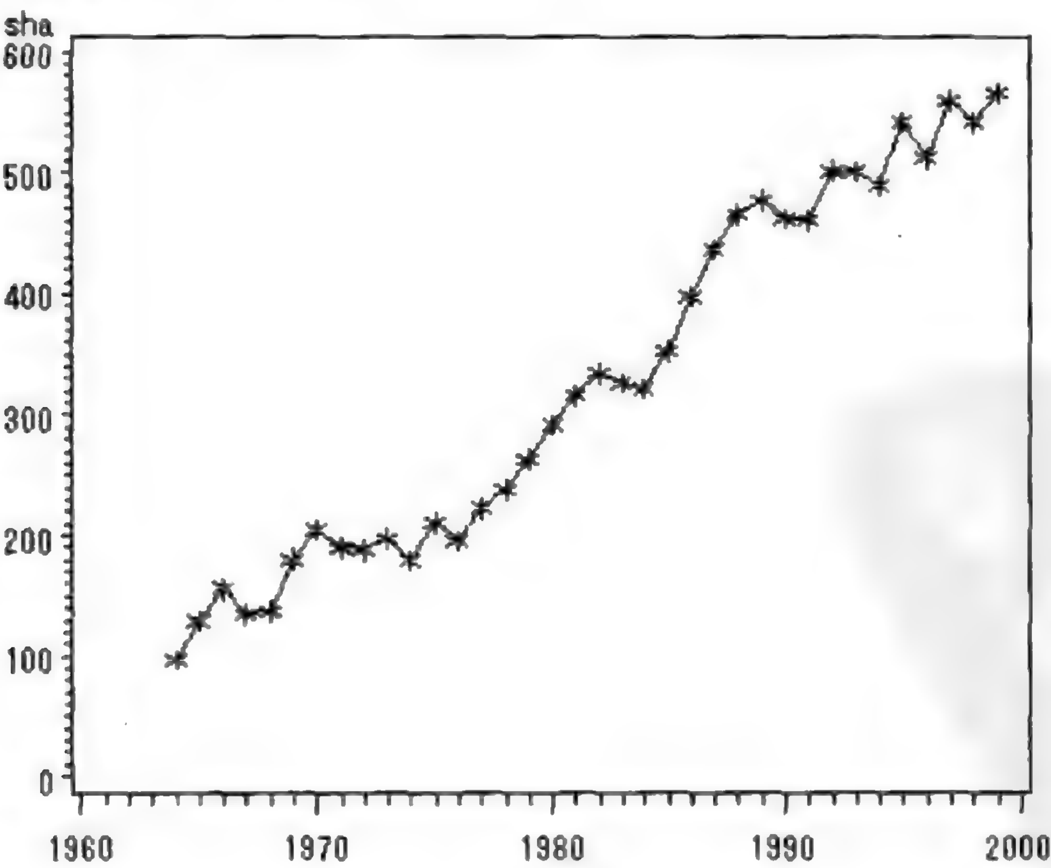


图 2.18 原序列时序图

从图 2.18 可知，中国纱年产量序列蕴含着显著的线性上升趋势，不是平稳序列. 原序列图有显著的线性趋势，通过一阶差分就可以实现趋势平稳. 一阶差分后得到差分后序列时序图，如图 2.19 所示.

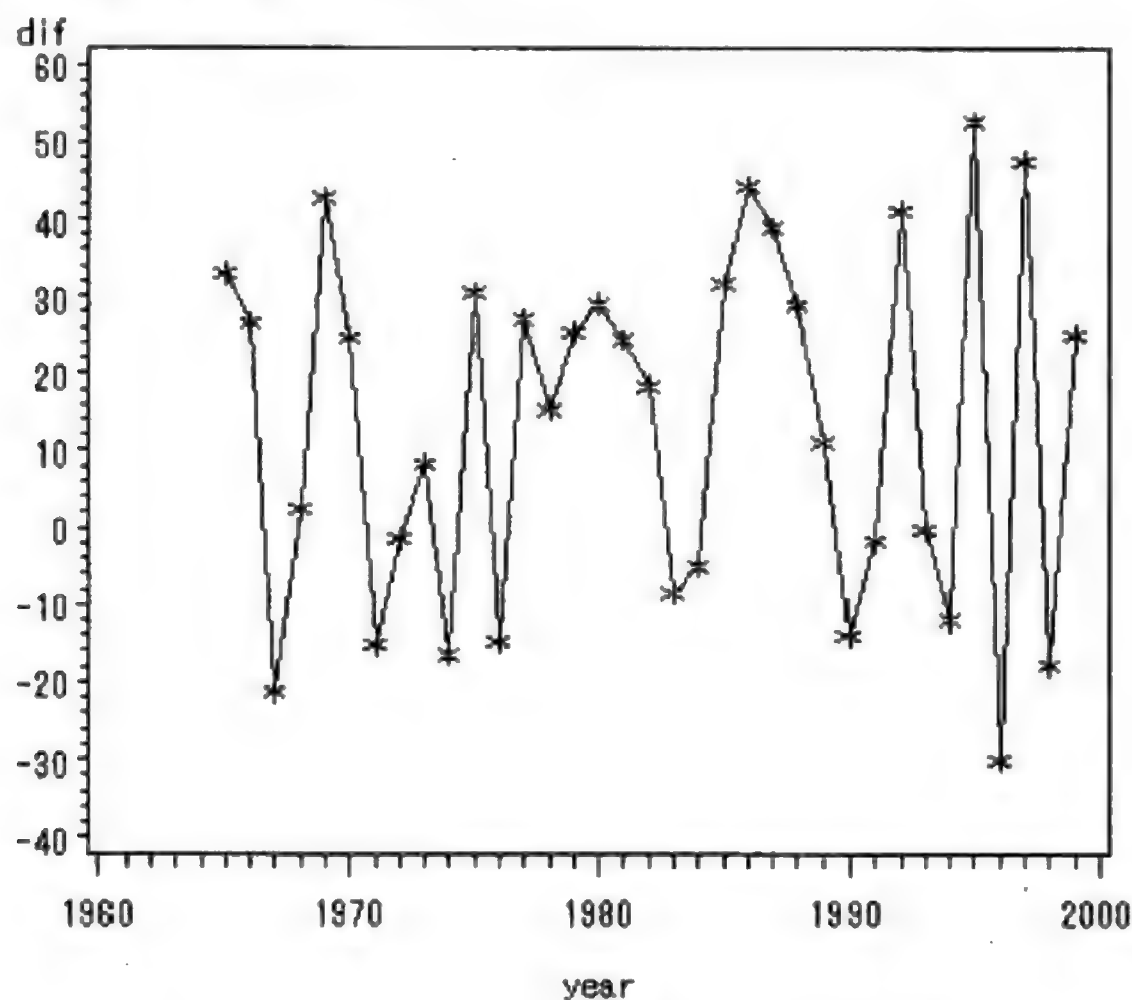


图 2.19　一阶差分后得到差分后序列时序图

从图 2.19 可知，该一阶差分后得到差分后序列的时序图始终在一个常数值附近随机波动，而且波动的范围有界、无明显趋势及周期特征，呈现出平稳的随机波动. 可见，一阶差分已经成功地从原序列中提取出线性趋势.

(2) 序列蕴含着曲线趋势，通常低阶(二阶或三阶)差分就可以提取出曲线趋势的影响.

例 2.13　尝试提取 1950～1999 年北京市民用车辆拥有量序列的确定性信息.

分析　北京市民用车辆拥有量序列蕴含着曲线趋势，通常低阶(二阶或三阶)差分就可以提取出曲线趋势的影响.

SAS 程序

```
data a;
input year x;
dif1=dif(x);  /*一阶差分*/
dif2=dif(dif1);  /*二阶差分*/
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
```

```
;
proc gplot;
plot x*year;plot dif1*year;plot dif2*year;
symbol v=star c=red i=join;
run;
```

原序列时序图如图 2.20 所示.

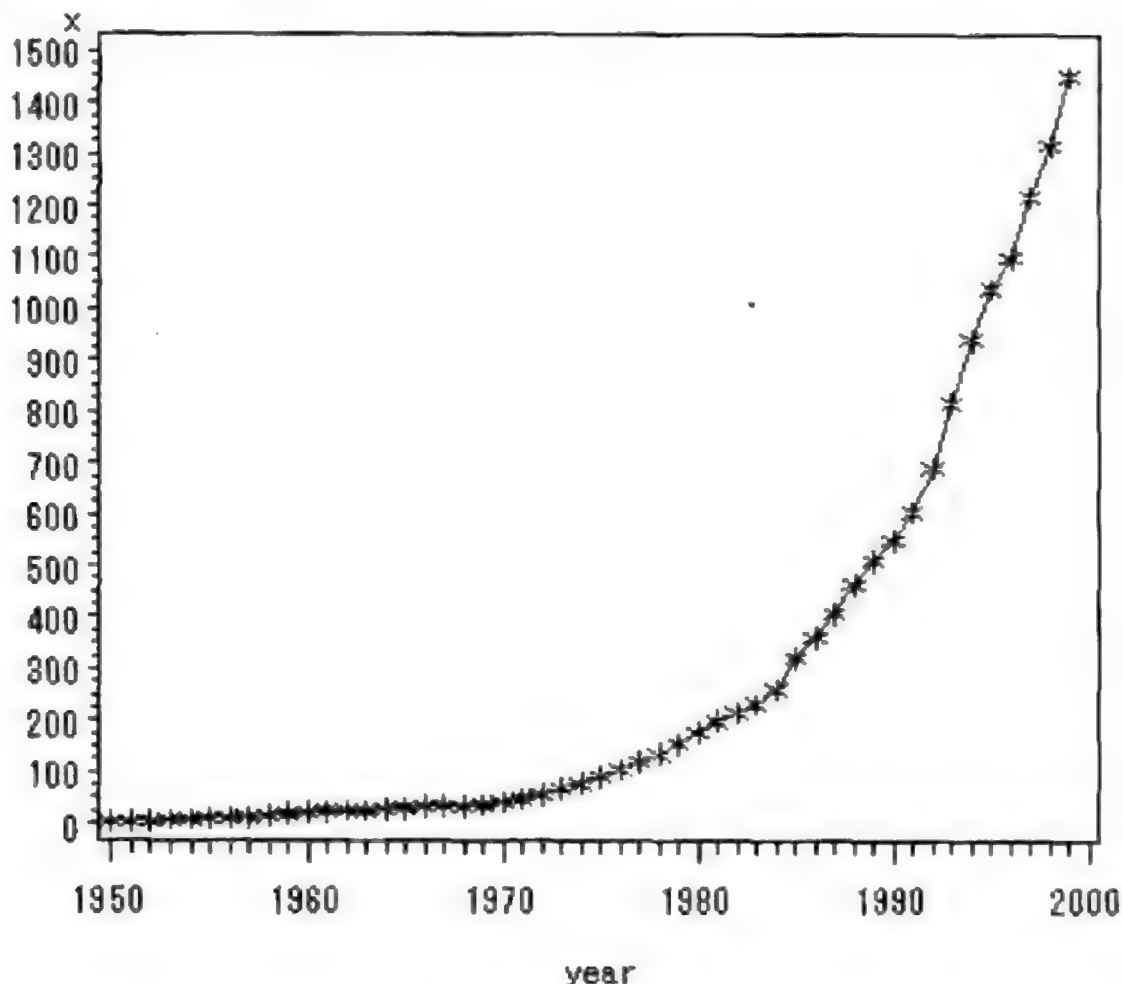


图 2.20　原序列时序图

从图 2.20 可知,北京市民用车辆拥有量序列蕴含着显著的曲线趋势,不是平稳序列,通常低阶(二阶或三阶)差分就可以提取出曲线趋势的影响. 一阶差分后得到差分后序列时序图,如图 2.21 所示.

从图 2.21 可知,一阶差分提取了原序列中部分长期趋势,但长期的趋势信息提取不充分,于是对一阶差分后序列再做一次差分运算,得到图 2.22.

从图 2.22 可知,二阶差分比较充分地提取了原序列中蕴含的长期信息,使得差分后序列不再呈示确定性趋势.

(3) 对于蕴含着固定周期的序列进行步长为周期长度的差分运算,通常可以较好地提取周期信息.

例 2.14　差分运算提取 1962 年 1 月～1975 年 12 月平均每头奶牛的月产奶量序列中的确定性信息.

分析　平均每头奶牛的月产奶量序列中蕴含着固定周期,可以对其进行步长为

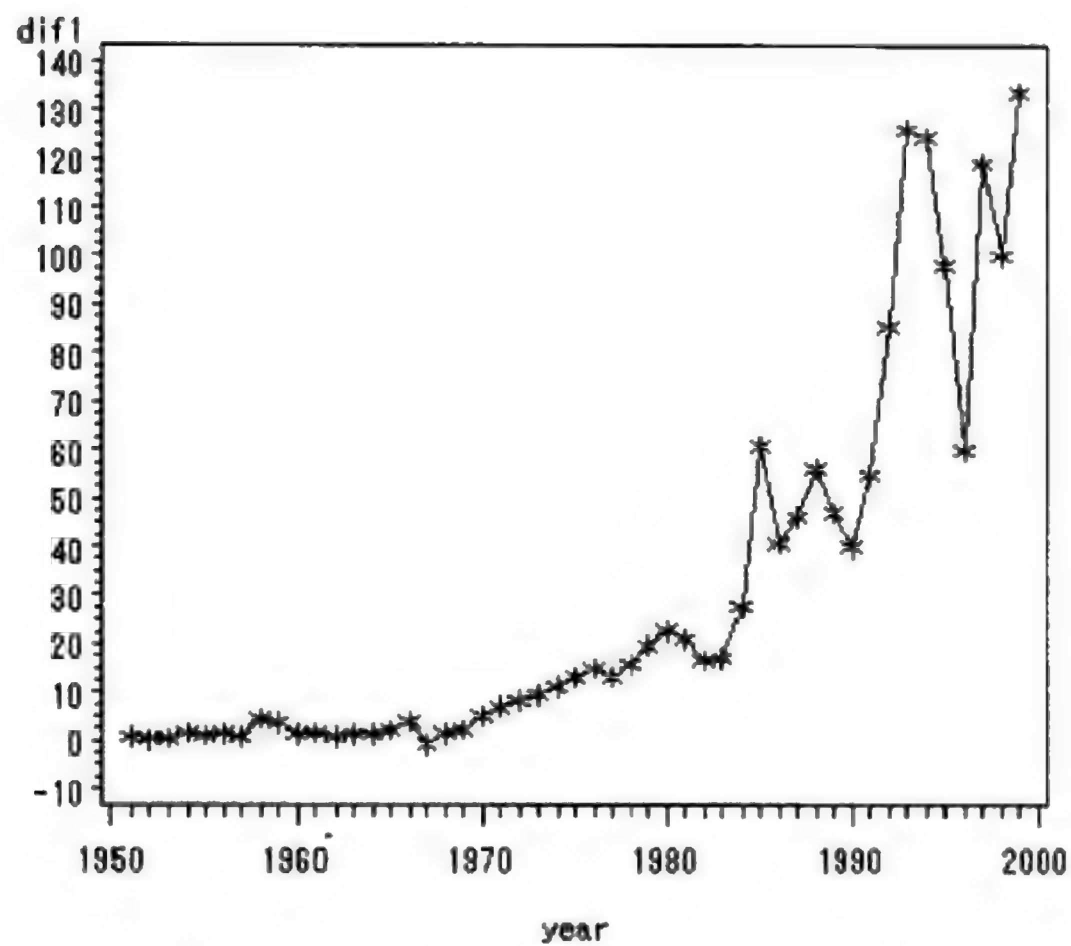


图 2.21 一阶差分后得到差分后序列时序图

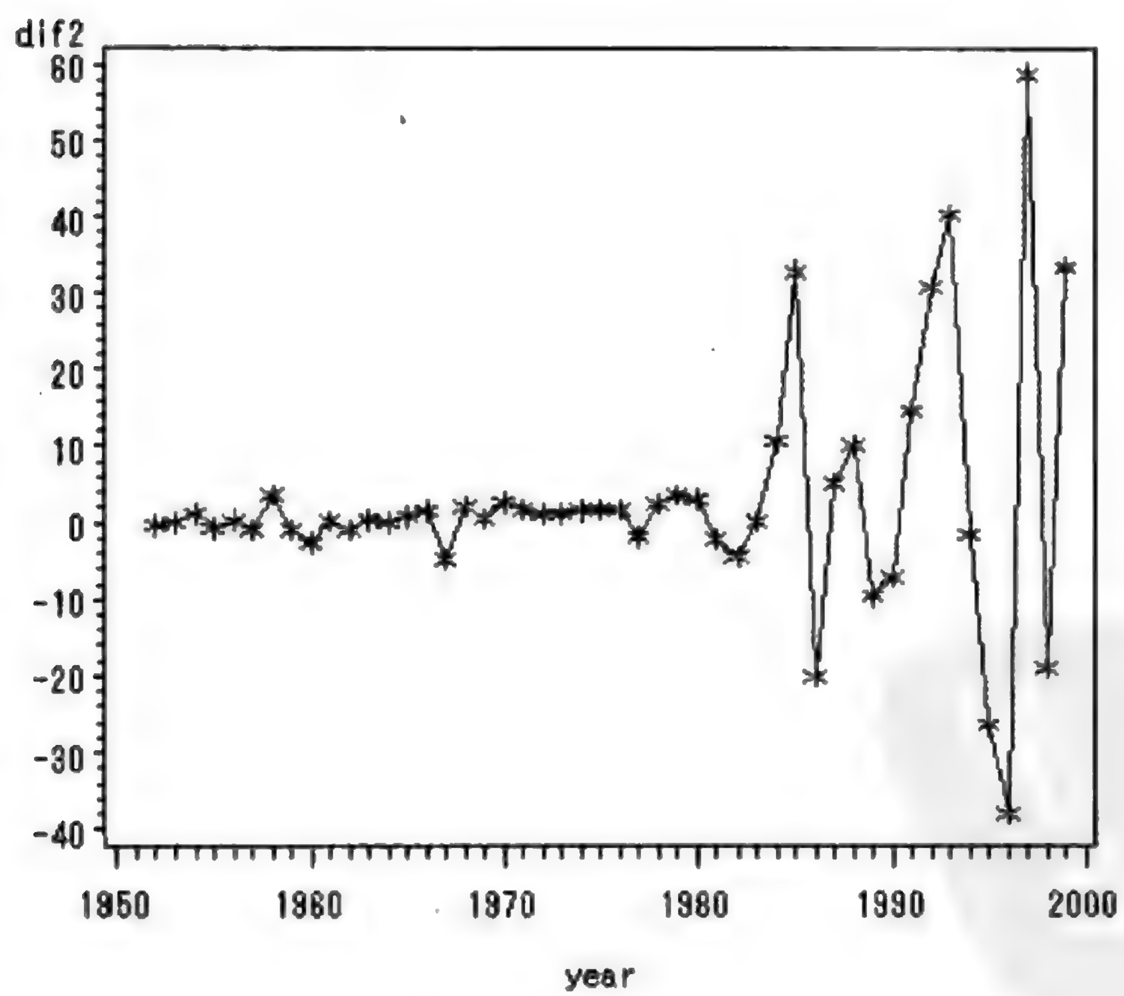


图 2.22 二阶差分后得到差分后序列时序图

周期长度的差分运算提取确定性信息.

SAS 程序

```
data a;
input milk@@;
time=intnx('month','1jan1962'd,_n_-1);
format time year4.;
dif1=dif(milk);
dif1_12=dif12(dif1);  /*一阶 12 步差分*/
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
;
proc gplot;
plot milk*time dif1*time dif1_12*time;
symbol v=diamond c=blue i=join;
run;
```

原序列时序图如图 2.23 所示.

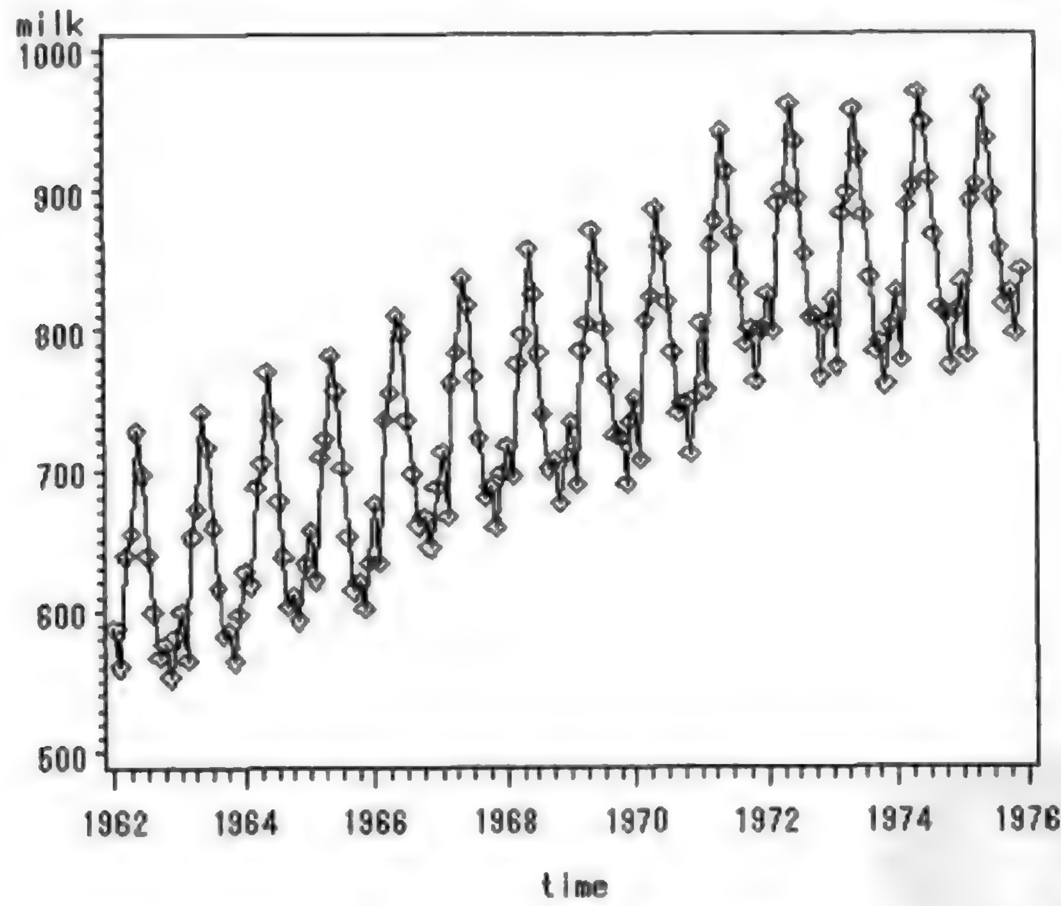


图 2.23 原序列时序图

该时序图显示该序列具有一个线性增长的趋势和周期一年的稳定季节变动,先对其做一步差分,提取线性递增趋势. 一阶差分后得到差分后序列时序图,如图 2.24 所示.

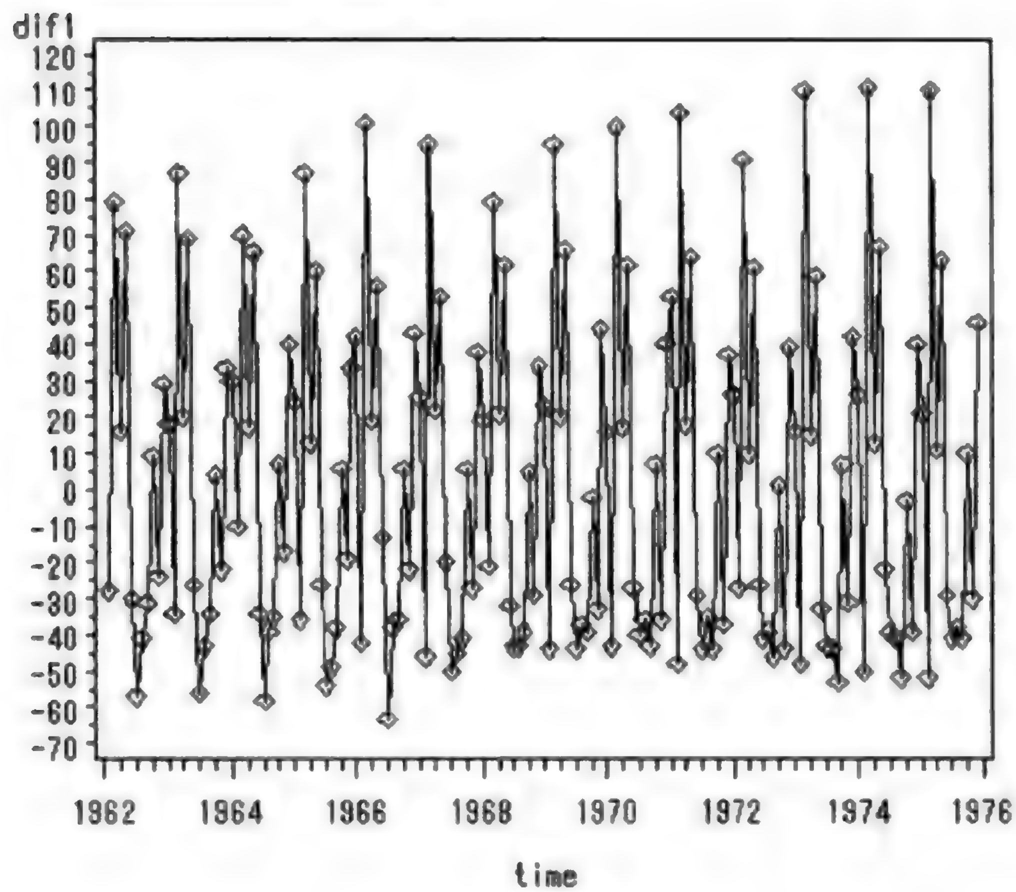


图 2.24　一阶差分后时序图

图 2.24 显示一阶差分后线性递增趋势已被提取，下面继续进行 12 步差分，提取季节波动信息. 12 步差分后的时序图如图 2.25 所示.

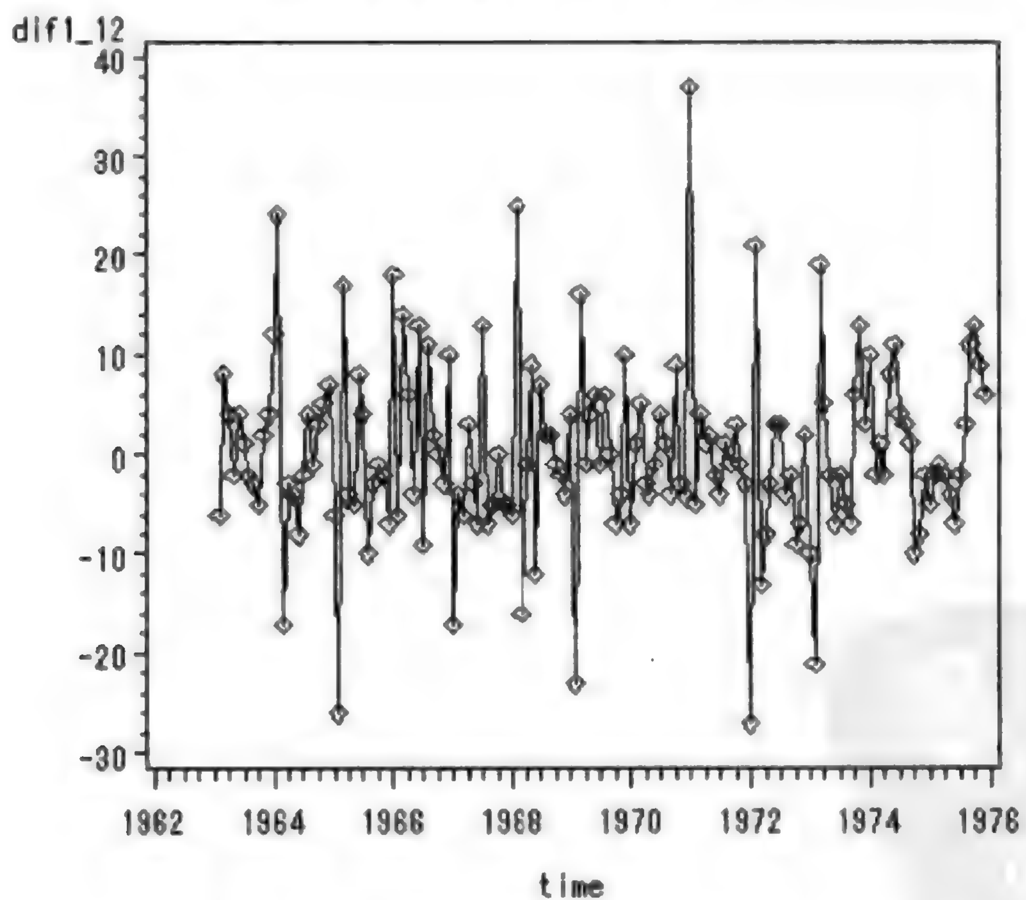


图 2.25　一阶 12 步差分后时序图

从图 2.25 可知，平均每头奶牛的月产奶量序列中递增和固定周期的信息已经被提取.

3. 过差分

从理论上来说,足够多次的差分运算可以充分地提取原序列中的非平稳确定性信息,但是,过度的差分会造成有用信息的浪费. 因此,在实际运用中差分运算阶数应当要适当,避免过度差分.

2.3.2 ARIMA 模型

差分运算具有强大的确定信息提取能力,对差分运算后得到的平稳序列可用 ARIMA 模型进行拟合.

1. ARIMA 模型结构

ARIMA 模型使用场合:差分平稳序列拟合.

具有如下结构的模型称为 ARIMA(p,d,q)模型:

$$\begin{cases}\Phi(B)\nabla^d x_t=\Theta(B)\varepsilon_t\\ E(\varepsilon_t)=0,\mathrm{Var}(\varepsilon_t)=\sigma_\varepsilon^2,E(\varepsilon_t\varepsilon_s)=0,s\neq t\\ E(x_s\varepsilon_t)=0,\forall s<t\end{cases}$$

特别地,当 $d=0$ 时,ARIMA(p,d,q)模型实际上就是 ARMA(p,q)模型;

当 $p=0$ 时,ARIMA(p,d,q)模型实际上就是 IMA(d,q)模型;

当 $q=0$ 时,ARIMA(p,d,q)模型实际上就是 ARI(d,q)模型;

当 $d=1,p=q=0$ 时,ARIMA(p,d,q)模型记为

$$\begin{cases}x_t=x_{t-1}+\varepsilon_t\\ E(\varepsilon_t)=0,\mathrm{Var}(\varepsilon_t)=\sigma_\varepsilon^2,E(\varepsilon_t\varepsilon_s)=0,s\neq t\\ E(x_s\varepsilon_t)=0,\forall s<t\end{cases}$$

该模型称为随机游走模型.

2. ARIMA 模型建模

ARIMA 模型建模与 ARMA 模型建模过程类似,其流程图如图 2.26 所示.

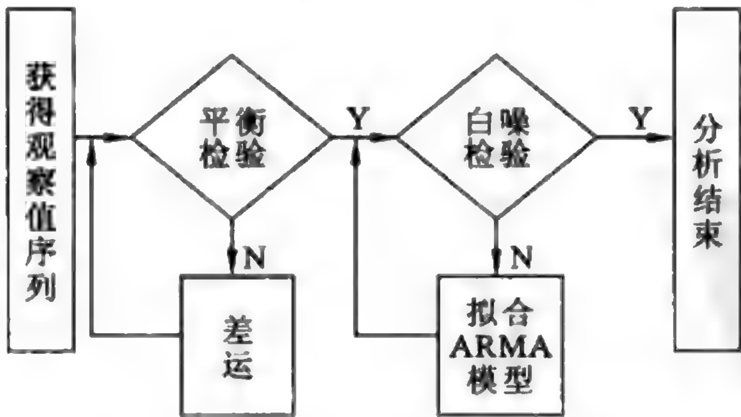


图 2.26 ARIMA 模型建模流程图

读者可根据例题来进一步熟悉 ARIMA 模型建模.

3. 疏系数模型

ARIMA(p,d,q)模型是指 d 阶差分后自相关最高阶数为 p,移动平均最高阶数

为 q 的模型，通常它包含 $p+q$ 个独立的未知系数：$\phi_1,\phi_2,\cdots,\phi_p,\theta_1,\theta_2,\cdots,\theta_q$.

如果该模型中有部分自相关系数 $\phi_j(1\leqslant j<p)$ 或部分移动平滑系数 $\theta_k(1\leqslant k<q)$ 为零，即原 ARIMA(p,d,q) 模型中有部分系数省缺了，那么该模型称为疏系数模型.

如果只是自相关部分有省缺系数，那么该疏系数模型可以简记为

$$\text{ARIMA}((p_1,p_2,\cdots,p_m),d,q)$$

其中，$p_1,p_2,\cdots,p_m$ 为非零自相关系数的阶数.

如果只是移动平滑部分有省缺系数，那么该疏系数模型可以简记为

$$\text{ARIMA}(p,d,(q_1,q_2,\cdots,q_n))$$

其中，$p_1,\cdots,p_m$ 为非零自相关系数的阶数.

如果自相关和移动平均部分都有省缺，可以简记为

$$\text{ARIMA}((p_1,p_2,\cdots,p_m),d,(q_1,q_2,\cdots,q_n))$$

读者可根据例题来进一步熟悉疏系数模型.

4. 季节模型

根据季节效应提取的难易程度，可以分为简单季节模型和乘积季节模型.

1）简单季节模型

简单季节模型是指序列中的季节效应和其他效应之间是加法关系：

$$x_t=S_t+T_t+I_t$$

简单季节模型通过简单的趋势差分、季节差分之后序列即可转化为平稳，它的模型结构通常如下(季节类似周期性)：

$$\nabla_D\nabla^d x_t=\frac{\Theta(B)}{\Phi(B)}\varepsilon_t$$

2）乘积季节模型

乘积季节模型使用场合：序列的季节效应、长期趋势效应和随机波动之间有着复杂的相互关联性，简单的季节模型不能充分地提取其中的相关关系.

构造原理：短期相关性用低阶 ARMA(p,q) 模型提取，季节相关性用以周期步长 S 为单位的 ARMA(P,Q) 模型提取. 模型记为 ARIMA$(p,d,q)\times(P,D,Q,)_S$.

假设短期相关和季节效应之间具有乘积关系，模型结构如下：

$$\nabla^d\nabla_S^D x_t=\frac{\Theta(B)}{\Phi(B)}\frac{\Theta_S(B)}{\Phi_S(B)}\varepsilon_t$$

以上详细理论见书后参考文献.

读者可根据例题来进一步熟悉季节模型.

例 2.15 （简单季节模型）拟合 1962～1991 年某国工人季度失业率序列.

分析 某国工人季度失业率序列蕴含着显著的线性趋势，一阶差分就可以实现趋势平稳. 再作 4 步差分消除季节效应的影响，该序列平稳之后才能对其做其他一系列处理.

SAS 程序

```
data a;
input x@@;
```

```
dif1_4=dif4(dif(x));  /*阶 4 步差分*/
time=intnx('quarter','1jan1962'd,_n_-1);
format time year4.;
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
;
proc gplot;
plot x*time dif1_4*time;
symbol c=black i=join v=star;
proc arima;
identify var=x(1,4);  /*一阶 4 步差分*/
estimate p=(1 4) noint;  /*疏系数,只有 p=1,4,无常数*/
forecast lead=0 id=time out=out;  /*forecast 为预测关键词,lead 为预测阶数*/
proc gplot data=out;
plot x*time=1 forecast*time=2 /overlay;
symbol1 c=black i=none v=star;
symbol2 c=red i=join v=none;
run;
```

对原序列作一阶差分消除趋势,再作 4 步差分消除季节效应的影响,差分后序列的时序图,如图 2.27 所示.

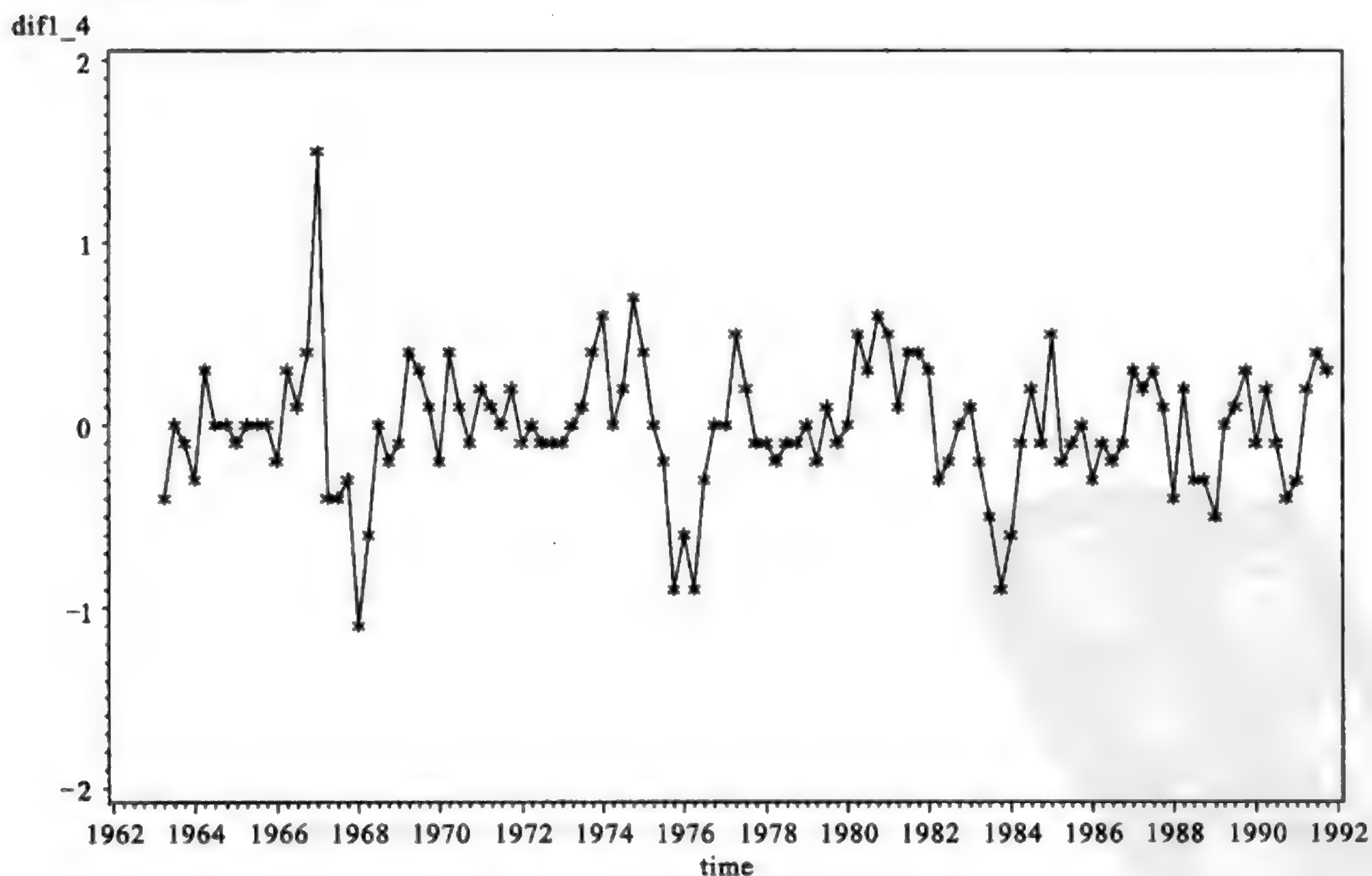


图 2.27　差分后序列的时序图

得到差分后的序列,该序列显示序列波动比较平稳. 其次给出白噪声检验结果如下:

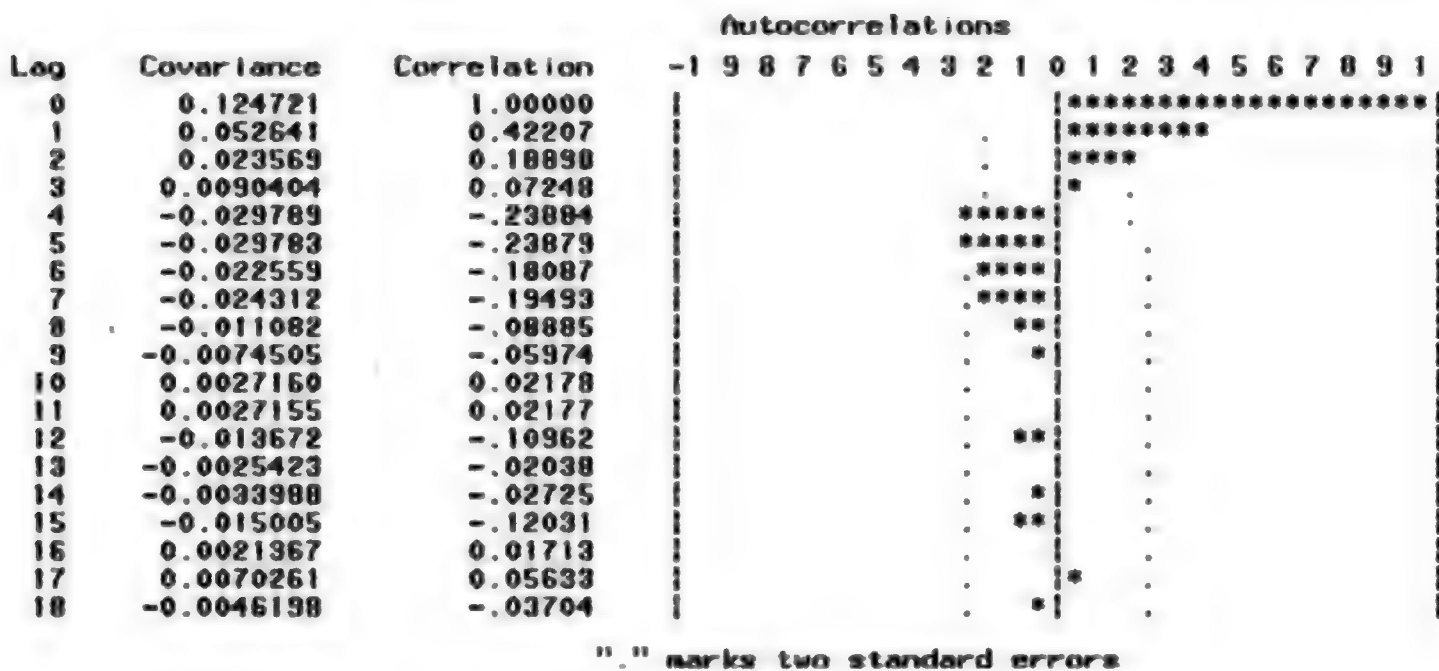
The ARIMA Procedure

Autocorrelation Check for White Noise

To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	43.84	6	<.0001	0.422	0.189	0.072	-0.239	-0.239	-0.181
12	51.71	12	<.0001	-0.195	-0.089	-0.060	0.022	0.022	-0.110
18	54.48	18	<.0001	-0.020	-0.027	-0.120	0.017	0.056	-0.037
24	59.44	24	<.0001	0.033	-0.057	-0.049	-0.007	-0.115	-0.118

以上结果显示,由于统计量的 P 值小于 0.0001,则 P 值远远小于 0.05,则认为在 0.05 显著水平下拒绝原假设,即拒绝序列为纯随机序列的假定.

下面观察差分后序列自相关图、偏自相关图来确定模型,结果如图 2.28 所示. 差分后序列偏自相关图如图 2.29 所示.

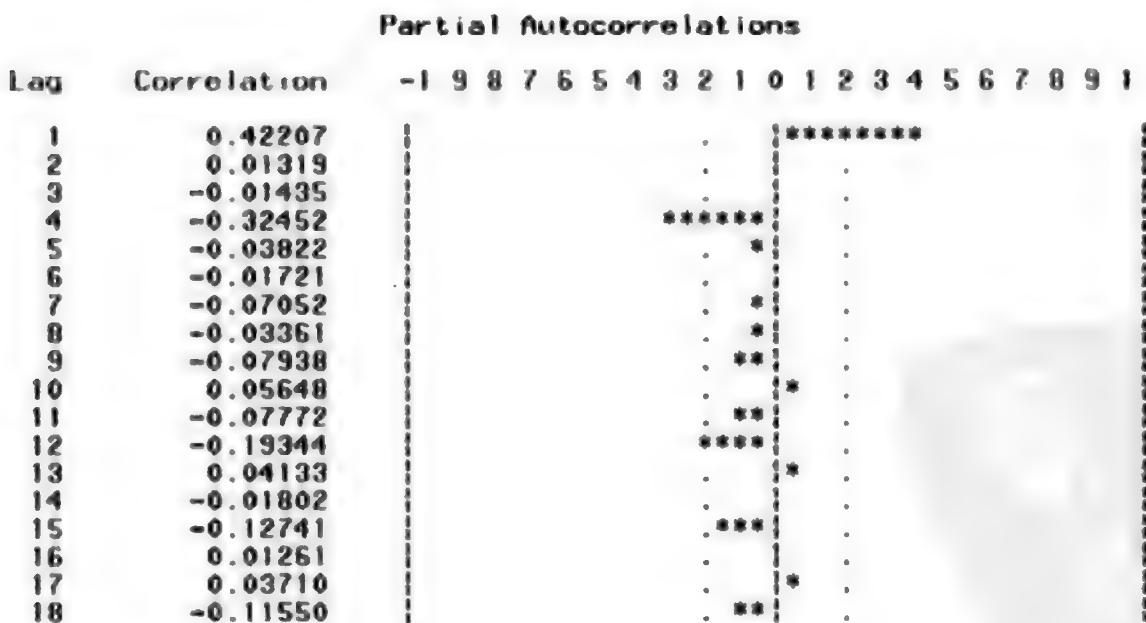


Autocorrelations

Lag	Covariance	Correlation
0	0.124721	1.00000
1	0.052641	0.42207
2	0.023569	0.18898
3	0.0090404	0.07248
4	-0.029789	-.23884
5	-0.029783	-.23879
6	-0.022559	-.18087
7	-0.024312	-.19493
8	-0.011082	-.08885
9	-0.0074505	-.05974
10	0.0027160	0.02178
11	0.0027155	0.02177
12	-0.013672	-.10962
13	-0.0025423	-.02038
14	-0.0033988	-.02725
15	-0.015005	-.12031
16	0.0021367	0.01713
17	0.0070261	0.05633
18	-0.0046198	-.03704

"." marks two standard errors

图 2.28　差分后自相关序列的时序图



Partial Autocorrelations

Lag	Correlation
1	0.42207
2	0.01319
3	-0.01435
4	-0.32452
5	-0.03822
6	-0.01721
7	-0.07052
8	-0.03361
9	-0.07938
10	0.05648
11	-0.07772
12	-0.19344
13	0.04133
14	-0.01802
15	-0.12741
16	0.01261
17	0.03710
18	-0.11550

图 2.29　差分后序列偏自相关图

综合以上分析,我们可以定模型为 ARIMA((1,4),1,0)模型. 模型得到的方程为(程序结果略去).

$$(1-B)(1-B^4)x_t=\frac{1}{1-0.44746B+0.28132B^4}\varepsilon_t$$

对模型的残差进行白噪声检验，结果表明残差为白噪声序列，表明模型达到要求；对参数进行显著性检验，结果表明均通过检验. 具体程序结果均略.

例 2.16 （乘积季节模型）拟合 1948～1981 年某国工人月度失业率序列.

分析 某国工人月度失业率序列蕴含着显著的线性趋势，一阶差分就可以实现趋势平稳. 再作 12 步差分消除季节效应的影响，该序列平稳之后才能对其做其他一系列处理.

SAS 程序

```
data a;
input x@@;
dif1_12=dif12(dif(x));
time=intnx('month','1jan1948'd,_n_-1);
format time year4.;
cards;
/*数据省略,见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目*/
;
proc gplot;
plot x*time dif1_12*time;
symbol c=black i=join v=none;
proc arima;
identify var=x(1,12);
estimate q=(1)(12) p=1 noint;  /*表示 s=12,D=1,p=1,q=1,P=0,Q=1*/
forecast lead=0 id=time out=out;
proc gplot data=out;
plot x*time=1 forecast*time=2/overlay;
symbol1 c=black i=none v=star;
symbol2 c=red i=join v=none;
run;
```

对原序列作一阶差分消除趋势，再作 12 步差分消除季节效应的影响，差分后序列的时序图如图 2.30 所示.

得到差分后的序列，该序列显示序列波动比较平稳. 其次给出白噪声检验结果如下：

以上结果显示，由于统计量的 P 值小于 0.0001，则 P 值远远小于 α ($\alpha=0.05$)，则认为在 $1-\alpha$ 的置信水平下拒绝原假设，即拒绝序列为纯随机序列的假定.

下面观察差分后序列自相关图、偏自相关图来确定模型，结果如图 2.31 所示. 差分后序列偏自相关图如图 2.32 所示.

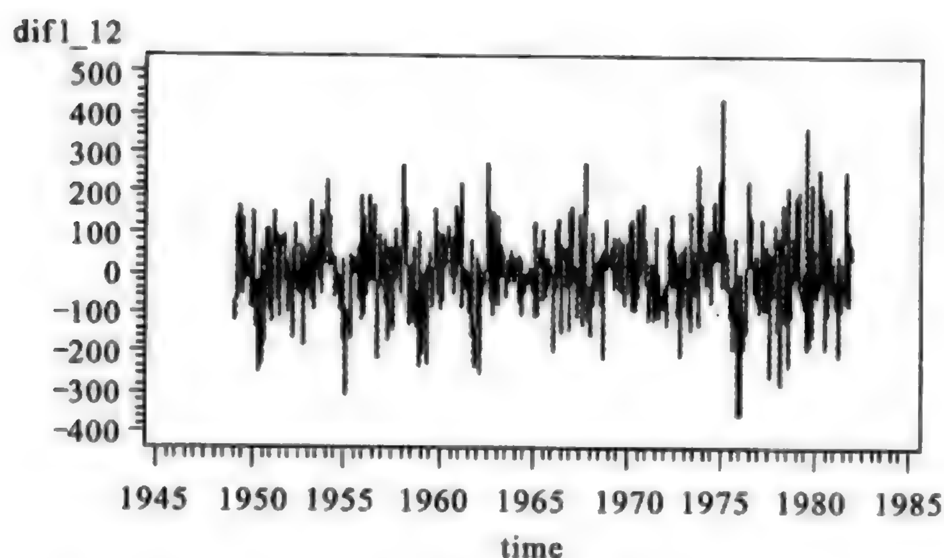


图 2.30　美国女性月度失业率差分后序列的时序图

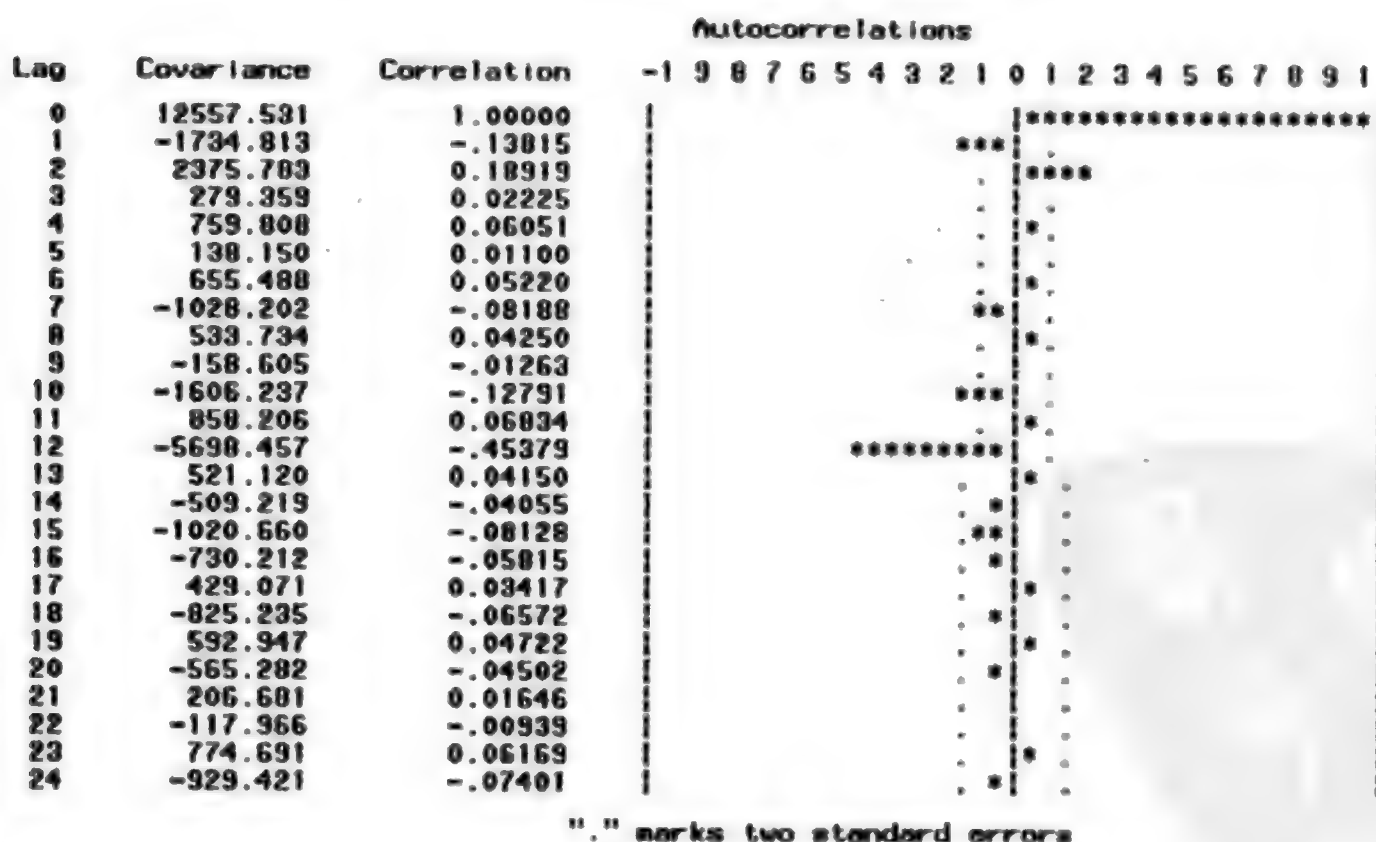
The ARIMA Procedure

Autocorrelation Check for White Noise

To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	43.84	6	<.0001	0.422	0.189	0.072	-0.239	-0.238	-0.181
12	51.71	12	<.0001	-0.195	-0.089	-0.060	0.022	0.022	-0.110
18	54.48	18	<.0001	-0.020	-0.027	-0.120	0.017	0.056	-0.037
24	59.44	24	<.0001	0.033	-0.057	-0.049	-0.007	-0.115	-0.118

综合以上分析，可以判定模型为 ARIMA$(1,1,1)\times(0,1,1)_{12}$模型. 模型得到的方程为

$$\nabla\nabla_{12}x_t=\frac{1+0.66137B}{1+0.78978B}(1-0.77394B^{12})\varepsilon_t$$

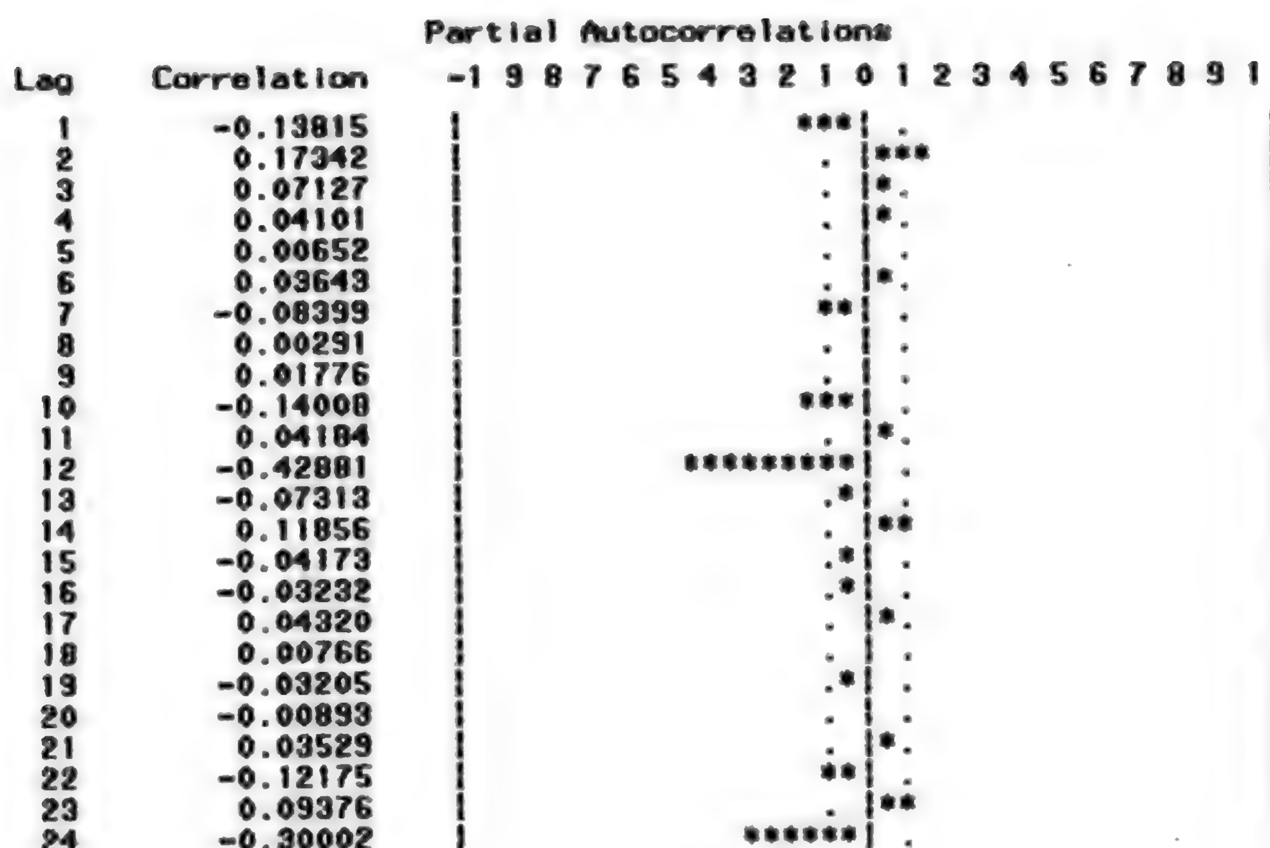


Autocorrelations

Lag	Covariance	Correlation
0	12557.531	1.00000
1	-1734.813	-.13815
2	2375.783	0.18919
3	279.359	0.02225
4	759.808	0.06051
5	138.150	0.01100
6	655.488	0.05220
7	-1028.202	-.08188
8	533.734	0.04250
9	-158.605	-.01263
10	-1606.237	-.12791
11	858.206	0.06834
12	-5698.457	-.45379
13	521.120	0.04150
14	-509.219	-.04055
15	-1020.660	-.08128
16	-730.212	-.05815
17	429.071	0.03417
18	-825.235	-.06572
19	592.947	0.04722
20	-565.282	-.04502
21	206.681	0.01646
22	-117.966	-.00939
23	774.691	0.06169
24	-929.421	-.07401

"." marks two standard errors

图 2.31　差分后自相关图



Partial Autocorrelations

Lag	Correlation
1	-0.13815
2	0.17342
3	0.07127
4	0.04101
5	0.00652
6	0.03643
7	-0.08399
8	0.00291
9	0.01776
10	-0.14008
11	0.04184
12	-0.42881
13	-0.07313
14	0.11856
15	-0.04173
16	-0.03232
17	0.04320
18	0.00766
19	-0.03205
20	-0.00893
21	0.03529
22	-0.12175
23	0.09376
24	-0.30002

图 2.32　差分后偏自相关图

对模型的残差进行白噪声检验,结果如下:

Autocorrelation Check of Residuals

To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	2.08	4	0.7191	-0.044	0.008	0.111	-0.018	-0.049	0.024
12	10.99	10	0.3584	-0.090	-0.100	-0.107	0.020	-0.002	-0.198
18	16.38	16	0.4265	0.036	0.016	-0.139	-0.002	0.096	-0.098
24	20.38	22	0.5593	0.027	-0.112	-0.017	0.043	-0.070	-0.086

结果表明残差均为白噪声序列,即模型没有损失太多信息,模型达到精度要求.

再对参数进行检验,结果如下:

Conditional Least Squares Estimation

Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
AR1,1	0.44746	0.08166	5.48	<.0001	1
AR1,2	-0.28132	0.08251	-3.41	0.0009	4

以上结果显示,由于统计量的 P 值远小于 0.05,所以参数都通过了检验.

最后对模型进行拟合图形,如图 2.33 所示.

图 2.33 中曲线表示拟合曲线,黑色点表示实际点. 从图像中可以很直观地发现其拟合效果还是很好的.

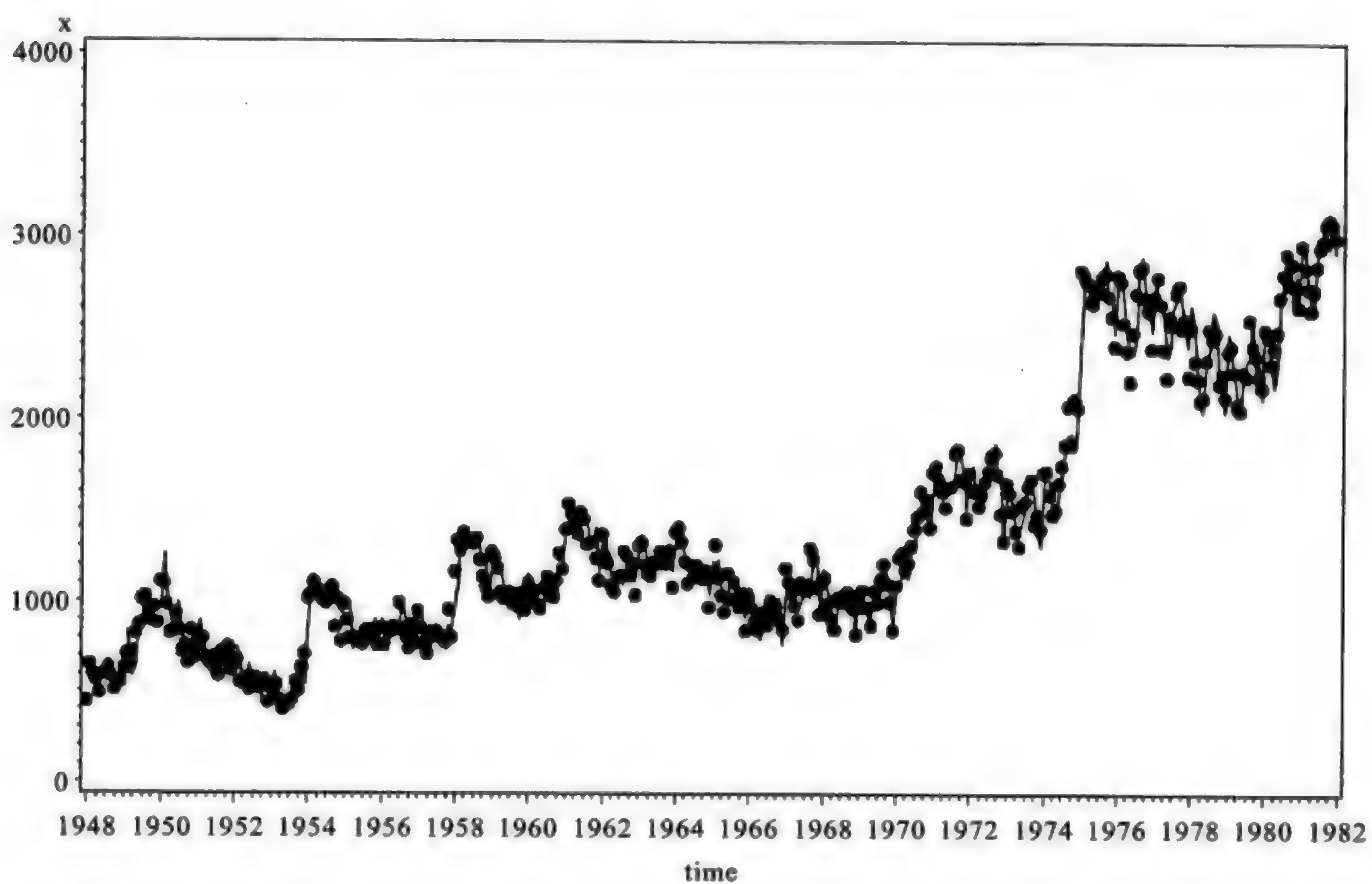


图 2.33 模型拟合效果图

第3章 数学规划

实际问题中，人们经常遇到这样一类决策问题：在一系列客观或主观限制条件下，寻求使所关注的某个或多个指标达到最大（或最小）的决策. 这种决策问题通常称为优化问题. 解决这类问题的方法称为最优化方法，又称为数学规划，它是运筹学里一个十分重要的分支.

最优化问题的数学模型的一般形式为

$$\text{opt } z=f(x) \tag{1}$$

$$\text{s.t.}\begin{cases} h_i(x)=0, & i=1,2,\cdots,l \\ g_j(x)\leqslant 0, & j=1,2,\cdots,m \\ t_k(x)\geqslant 0, & k=1,2,\cdots,n \\ x\in D\subseteq \mathbf{R}^n \end{cases} \tag{2}$$

opt(optimize)是最优化的意思，可以是求最小 min(minimize)或求最大 max (maximize)，s. t. (subject to)是“受约束于”.

模型包含三个要素：决策变量(decision variable)、目标函数(objective function)和约束条件(constraints). (2)式所确定的 x 的范围称为可行域(feasible region)，满足(2)式的解 x 称为可行解(feasible solution)，同时满足(1)、(2)式的解 x^* 称为最优解(optimal solution)，整个可行域上的最优解称为全局最优解(global optimal solution)，可行域中某个领域上的最优解称为局部最优解(local optimal solution). 最优解所对应的目标函数值称为最优值(optimum).

不同优化模型的求解方法以及求解难度有很大的不同，可按如下方法对模型进行分类：

(1) 按有无约束条件可分为：①无约束优化(unconstrained optimization)，这类问题蕴含了重要的寻优计算方法；②约束优化(constrained optimization)，大部分实际问题都是约束优化问题.

(2) 按决策变量取值是否连续可分为：

① 数学规划(mathematical programming)或连续优化(continuous optmization). 可继续划分为线性规划(linear programming，简称 LP)和非线性规划(nonlinear programming，简称 NLP). 在非线性规划中有一种规划称为二次规划(quadratic programming，简称 QP)，二次规划问题的目标为二次函数，约束为线性函数.

② 离散优化(discrete optimization)或组合优化(combinatorial optimization). 这类优化问题中包含一种常用的优化——整数规划(integer programming，简称 IP)，整数规

划中又包含很重要的一类规划——0-1(整数)规划(zero-one programming),这类规划问题的决策变量只取 0 或者 1.

在求解组合优化问题中,出现了很多现代优化计算方法.

(3) 按目标的多少可分为单目标规划和多目标规划.

(4) 按模型中参数和变量是否具有不确定性可分为确定性规划和不确定性规划.

(5) 按问题求解的特性可分为目标规划、动态规划、多层规划、网络优化等.

在本章中,我们将重点介绍线性规划、非线性规划、多目标规划和目标规划的模型建立及求解方法.

3.1 线性规划

线性规划一般形式: $\min(\text{或} \max) z = \sum_{j=1}^{n} c_j x_j$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_{ij} x_j = (\text{或} \leqslant \text{或} \geqslant) b_i, \quad i = 1, 2, \cdots n$$

$$x_j \geqslant 0, \quad j = 1, 2, \cdots, n$$

决策变量为 x_j,其他都为常数. 一般线性规划问题都可以通过引入非负的松弛变量(slack variable)与非负的剩余变量(surplus variable)的方法化为标准形式(约束全是等约束).

线性规划问题的可行域是一个凸集(convex set)(任意两点的连线上的点都在区域内部,可以视为没有凹坑的凸多面体),所以最优解在凸多面体的某个顶点上达到.

求解方法:单纯形算法(simplex method).

实质上就是在保证可行(用最小比值法则)的前提下,先在可行解中取一个顶点,判断有没有达到最优,如果没有,就按照一定的规则[让正检验数(testing number)对应的变量入基(base)],旋转到比这个顶点更优的一个点,再判断有没有达到最优,这样迭代下去,直到达到最优(或判断为不可行,或判断为无界)为止. 所以单纯形法实质上是一种具有某种规则的搜索算法. 而 LINGO 软件就是采用单纯形法对线性规划问题进行求解的.

3.1.1 连续型线性规划

如果实际问题具有如下性质,我们可以考虑建立线性规划模型.

(1) 比例性:每个决策变量对目标函数以及右端项的贡献与该决策变量的取值成正比.

(2) 可加性：每个决策变量对目标函数以及右端项的贡献与其他决策变量的取值无关.

(3) 连续性：每个决策变量的取值都是连续的.

比例性与可加性保证了目标函数和约束条件对于决策变量的线性性，连续性则允许得到决策变量的实数最优解. 如果决策变量不连续，考虑建立离散的优化模型.

例 3.1 (阶段生产问题)某公司生产某产品，最大生产能力为 10 000 单位，每单位存储费 2 元，预定的销售量与单位成本见表 3.1.

表 3.1 每个月的单位成本与销售量

月份	单位成本/元	销售量
1	70	6000
2	71	7000
3	80	12000
4	76	6000

求一生产计划，使①满足需求；②不超过生产能力；③成本(生产成本与存储费之和)最低.

分析 这是一个多阶段生产计划问题，涉及多阶段存储，因为只需要制定这 4 个月的生产计划，所以不妨假定 1 月初无库存，4 月底销售完，当月生产的不作为当月的库存，也就是只有 2 月，2～3 月，3～4 月的库存，库存量无限制.

模型建立

(1) 决策变量.

所谓生产计划，就是每月生产多少，差什么就设什么，所以设 x_i 为第 i 月产量. 为了建立模型方便，将其他已知的量设为字母，设 d_i 为销售量，e_i 为存储费，c_i 为单位成本.

(2) 目标函数.

生产成本：$\sum_{j=1}^{4} c_j x_j$.

总库存费：$j \sim j+1$ 月的库存量(记为第 $j+1$ 月的库存量) 应该是 $1 \sim j$ 月的总产量减去 $1 \sim j$ 月的总销售量，即 $\sum_{i=1}^{j} x_i - \sum_{i=1}^{j} d_i$. 乘上单位存储费即库存费. 即 $\sum_{j=1}^{3}\left(\sum_{i=1}^{j} x_i - \sum_{i=1}^{j} d_i\right)e_{j+1}$.

总成本：$f = \sum_{j=1}^{4} c_j x_j + \sum_{j=1}^{3}\left(\sum_{i=1}^{j} x_i - \sum_{i=1}^{j} d_i\right)e_{j+1}$，求其最小值.

(3) 约束.

满足需求：如果每个月末都会有非负的存储量，显然是满足需求的，因而可用

约束

$$\sum_{i=1}^{j} x_i - \sum_{i=1}^{j} d_i \geqslant 0, \quad j = 1,2,3$$

4 个月的计划:4 个月的总产量等于总需求量 $\sum_{j=1}^{4} x_j = \sum_{j=1}^{4} d_j$.

产量限制:因为产量比较大,不限制变量的取整数约束,视为一个连续变化的量.

综上所述,建立数学模型如下:

$$\min f = \sum_{j=1}^{4} c_j x_j + \sum_{j=1}^{3} \left(\sum_{i=1}^{j} x_i - \sum_{i=1}^{j} d_i \right) e_{j+1}$$

$$\text{s. t.} \begin{cases} \sum_{i=1}^{j} x_i \geqslant \sum_{i=1}^{j} d_i, & j = 1,2,3 \\ \sum_{i=1}^{4} x_i = \sum_{i=1}^{4} d_i \\ 0 \leqslant x_i \leqslant 10000, & i = 1,2,3,4 \end{cases}$$

LINGO 程序(字母不区分大小写)

```
model:  !程序开始;
title 生产计划程序 1;  ! 标题;
Sets:
yuefen/1..4/:c,x,e,d;  !设置集合;
endsets
data:  !输入数据;
c=70 71 80 76;
d=6000 7000 12000 6000;
e=2 2 2 2 ;
a=10000;
enddata
min=@sum(yuefen:c* x)+
             @sum(yuefen(j)|j#lt#4:
                  @sum(yuefen(i)|i#le#j:x-d)*e(j+1));  !目标函数;
@for(yuefen(j)|j#lt#4:
            @sum(yuefen(i)|i#le#j:x)>@sum(yuefen(i)|i#le#j:d));
             !1-3 月份需求约束;
@sum(yuefen:x)=@sum(yuefen:d);  !4 月份需求约束;
@for(yuefen:x<a);  !生产能力约束;
End  !程序结束;
```

注 (1) 集合定义部分. LINGO 将集合(SET)的概念引入建模语言,集合是一组

相关对象构成的组合，代表模型中的实际事物，并与数学变量联系起来，是实际问题到数学的抽象. 本例中 4 个月的生产量可以视为集合，4 个月的销量可以视为集合.

每个集合在使用前需要预先给出定义，定义集合时要明确三方面内容：集合的名称、集合内的成员（组成集合的个体，也称元素）以及集合的属性（可以看成是与集合有关的变量或常量，相当于数组），本例定义月产量集合：

```
yuefen/1..4/:x;
```

相同定义方式为

```
yuefen/1 2 3 4/: x;  或  yuefen/yuefen1..yuefen4/: x;
```

具体使用方式可根据需要选择. 集合定义部分语句以 sets：开始，以 endsets 结束，这两个语句必须单独成一行. endsets 后不加标点符号.

本程序中集合定义部分为

```
Sets:
yuefen/1..4/:c,x,e,d;  ! 设置集合;
endsets
```

(2) 数据初始化（数据段）. 集合定义段以 data：开始，以 enddata 结束. 这两个语句必须单独成行. 本程序中数据初始化段为

```
data:  !输入数据;
c=70 71 80 76;
d=6000 7000 12000 6000;
e=2 2 2 2;
a=10000;
end data
```

(3) LINGO 模型以语句 model：开始，以 end 结束，这两个语句单独成一行. 完整的模型由集合定义、数据段、目标函数和约束条件等部分组成，这几个部分的先后次序无关紧要，“!”开头的是注释语句（可有可无）. title 是标题语句（可有可无）.

运行结果如下：

```
Model Title::生产计划程序 1

     Variable          Value        Reduced Cost
        A            10000.00         0.000000
       C(1)          70.00000         0.000000
       C(2)          71.00000         0.000000
       C(3)          80.00000         0.000000
       C(4)          76.00000         0.000000
       X(1)          10000.00         0.000000
       X(2)          10000.00         0.000000
       X(3)          5000.000         0.000000
```

```
X(4)        6000.000        0.000000
E(1)        2.000000        0.000000
E(2)        2.000000        0.000000
E(3)        2.000000        0.000000
E(4)        2.000000        0.000000
D(1)        6000.000        0.000000
D(2)        7000.000        0.000000
D(3)        12000.00        0.000000
D(4)        6000.000        0.000000
```

结果分析 由运行结果 X 的值可知，1 月份生产 10000 件，2 月份生产 10000 件，3 月份生产 5000 件，4 月份生产 6000 件.

例 3.2 （生产决策问题）某工厂可以用 A，B 两种原料生产 I，II，III 三种产品（每种产品都同时需要用两种原料），有关数据见表 3.2.

表 3.2 单位消耗与资源限制

	产品 I	产品 II	产品 III	现有原料/t
原料 A	2	1	1	7
原料 B	1	3	2	11
单位产品利润/万元	2	3	1	

(1) 求最优的生产计划.

(2) 若产品 I 的单位利润上涨 1 万元，要不要改变生产计划？

(3) 若目前市场上原料 A 的实际价格为 0.5 万元/t，工厂应如何决策？

(4) 若目前市场上原料 A 的实际价格为 0.8 万元/t，工厂应如何决策？

模型建立 设 x_1, x_2, x_3 分别表示产品 I，II，III 的生产量，问题的模型建立如下：

$$\max f = 2x_1 + 3x_2 + x_3$$

$$\text{s.t.}\begin{cases} 2x_1 + x_2 + 2x_3 \leqslant 7 \\ x_1 + 3x_2 + 2x_3 \leqslant 11 \\ x_1, x_2, x_3 \geqslant 0 \end{cases}$$

LINGO 程序

```
model:
title 生产决策问题;
[maxf]max=2*x1+3*x2+x3;  !目标函数;
[A]2*x1+x2+2*x3<7;  !A 原料约束;
[B]x1+3*x2+2*x3<11;  !B 原料约束;
END
```

运行结果如下：

```
Model Title:生产决策问题

        Variable        Value        Reduced Cost
           X1          2.000000       0.000000
           X2          3.000000       0.000000
           X3          0.000000       1.800000
          Row     Slack or Surplus    Dual Price
           MAXF       13.00000        1.000000
           A          0.000000        0.6000000
           B          0.000000        0.8000000
```

打开敏感性分析开关，即打开 LINGO|Options| General Solver(通用求解程序)选项卡，在 dual Computations(对偶计算)下选择 Prices and Ranges 保存设置，求解后再在程序窗口下点击 LINGO|Range，调出敏感性分析报告：

```
Ranges in which the basis is unchanged:

                    Objective Coefficient Ranges
                  Current          Allowable        Allowable
 Variable        Coefficient        Increase         Decrease
   X1             2.000000          4.000000         1.000000
   X2             3.000000          3.000000         2.000000
   X3             1.000000          1.800000         INFINITY

                        Righthand Side Ranges
  Row               Current         Allowable        Allowable
                      RHS            Increase         Decrease
   A              7.000000          13.00000         3.333333
   B              11.00000          10.00000         7.500000
```

对问题(1)，最优生产计划是生产 2 t 产品 I，3 t 产品 II，最大利润 13 万元.

对问题(2)，由敏感性分析报告可看到，在不改变最优解的前提下，产品 I 的单位利润可以上涨 4 万元，所以不用改变生产计划，最大利润增加 2 万元.

对于问题(3)，如果目前市场上的原料 A 的实际价格是 $0.5<0.6$(A 的影子价格)，因此应该购进原料 A，扩大生产能力，最大购进数量是 15 t，利润将会增加

$$(0.6-0.5)\times 15=1.5(\text{万元})$$

对于问题(4)，如果目前市场上的原料 A 的实际价格是 $0.8>0.6$(A 的影子价格)，因此应该出售部分原料将更为赚钱，最大出售量为 3.33 t，利润将会增加

$$(0.8-0.6)\times 3.33=0.66(\text{万元})$$

这里需要说明的是，上面说的最大购买量或者出售量是在保证当前的最优基不

变的前提下的最大量，因为最优基不变才能保证当前的影子价格有意义. 例如，最大购进量 15 t，究竟能不能再多购进一些呢？这个要具体问题具体分析，可能再购进会剩余，也可能超过 15 t 的量在一个新的最优基对应的新的影子价格下增长利润. 可以自己动手更改约束的右端项运行程序观察.

3.1.2 整数线性规划与 0-1 规划

在线性规划问题中，有些最优解可能是分数或小数，但对于某些具体问题，常常要求解答必须是整数. 例如，所求解是机器的台数、工作的人数或装货的车数等. 为了满足整数的要求，初看起来似乎只要把已得的非整数解舍入化整就可以了. 实际上化整后的数不见得是可行解和最优解，所以应该有特殊的方法来求解整数规划. 在整数规划中，如果所有变量都限制为整数，则称为纯整数规划；如果仅一部分变量限制为整数，则称为混合整数规划. 整数规划的一种特殊情形是 0-1 规划，它的变数仅限于 0 或 1.

整数规划的求解方法是分枝定界法，其思想是“分而治之”的策略，即逐次对解空间进行划分. 所谓分枝，就是不断去掉非整数最优解的区域对可行域进行划分；所谓定界，如果在某区域得到整数最优解，可以作为原问题最优解的下界（对极小化问题），其他区域上的最优解（无整数限制）如果比它大，就可以剪枝，从而避免完全枚举.

0-1 规划的求解方法有隐枚举法（穷举法）、分枝定界法.

例 3.3 （下料问题 1）现要做 100 套钢架，用长为 2.9 m，2.1 m 和 1.5 m 的元钢各一根，已知原料长 7.4 m，问如何下料，使用的原材料最省？

分析 之所以有这样的问题，是因为下料的方式有很多种，如何下料，就是每一种下料方式下了多少根钢材，显然这就是我们的决策变量.

合理的下料方式是剩余料头的长度不会超过最短原料需求（本题为 1.5 m），下料的所有的合理方式可用 LINGO 编写搜索算法全部搜索出来.

LINGO 程序

```
model:
title 搜索合理的下料方式;
sets:
ren/1..5/:;
!用一根原料可下各需求长度的最多根数定义元素个数,最多为 4,这里要定义 5;
long(ren,ren,ren):;  !有三种需求长度,定义三维数组;
endsets
data:  !搜索所有满足过滤条件的 i,j,k;
@text('D:\renxinglong.txt')=@writefor(long(i,j,k)|
!一种下料方式下料长度和不超过总长度;
```

```
                    (7.4-2.9*(i-1)-2.1*(j-1)-1.5*(k-1))#ge#0
                       #and#
                    !合理模式的余料小于最短需求;
                    (7.4-2.9*(i-1)-2.1*(j-1)-1.5*(k-1))#lt#@smin(2.9,2.1,1.5)
        !输出下料方式到文本文件 renxinglong.txt,我们需要的数是 0--4;
        :i-1,4*'',j-1,4*'',k-1,@newline(2));
  enddata
    end
```

求解得到 8 种下料方式,程序自动在 D 盘建立 renxinglong. txt 文件并写入结果,对每一种方式再简单计算一下余料得到表 3.3.

表 3.3　所有下料方式表

	I	II	III	IV	V	VI	VII	VIII
2.9 m	2	1	1	1	0	0	0	0
2.1 m	0	2	1	0	3	2	1	0
1.5 m	1	0	1	3	0	2	3	4
剩余料头	0.1 m	0.3 m	0.9 m	0 m	1.1 m	0.2 m	0.8 m	1.4 m

模型建立　设 x_i 表示按第 i 种方式下料的根数 ($i=1,2,\cdots,8$,则问题的线性规划模型为

$$\min f=0.1x_1+0.3x_2+0.9x_3+0x_4+1.1x_5+0.2x_6+0.8x_7+1.4x_8$$

$$\text{s.t.}\begin{cases}2x_1+x_2+x_3+x_4\geqslant 100\\ 2x_2+3x_3+3x_5+2x_6+x_7\geqslant 100\\ x_1+x_3+3x_4+2x_6+3x_7+4x_8\geqslant 100\\ x_i\geqslant 0,j=1,2,3,4,5,6,7,8;x_i\text{ 取整}\end{cases}$$

模型说明　关于目标函数的选取有两种,一种是剩余的余料最少,上面选取的就是这个目标,另一种就是所用的原料的根数最少 $\left(\min f=\sum_{i=1}^{8}x_i\right)$. 这两个目标是有区别的,余料最少时所用的原料不一定最少,因为有截出多余的满足需求的元钢并没有算作余料,可以根据实际情况选择这两种目标.

决策变量要限制取整数,是一个纯整数线性规划问题.

这种全方式设变量的模型只适合小型下料问题,大型下料问题或者对下料方式有限制的问题将不适合.

LINGO 程序

```
  model:
  title 下料问题;
  !定义集合段;
```

```
sets:
hang/1..3/:b;  !定义矩阵的行;
lin/1..8/:c,x;  !定义矩阵的列以及变量;
xishu(hang,lin):a;  !定义约束的系数矩阵;
endsets
data:
!数据可以直接粘贴表格;
a=
2    1    1    1    0    0    0    0
0    2    1    0    3    2    1    0
1    0    1    3    0    2    3    4
;
c=
0.1   0.3   0.9   0    1.1   0.2   0.8   1.4
;
!目标函数的系数;
b=100 100 100;  !约束的右端项;
enddata
[obj]min=@sum(lin:c*x);
@for(hang(I):[yueshu]
  @sum(lin(j):a(i,j)*x(j))>b(i));
@for(lie:@gin(x));  !限制变量取整数;
end
```

运行结果如下：

```
global optimal solution found.
  objective value:                          10.00000
  total solver iterations:                         0
  model title:下料问题
            variable          value          reduced cost
            ......
              X(1)          0.000000          0.1000000
              X(2)          0.000000          0.1000000
              X(3)          0.000000          0.8000000
              X(4)          100.0000           0.000000
              X(5)          0.000000          0.8000000
              X(6)          50.00000           0.000000
              X(7)          0.000000          0.7000000
              X(8)          0.000000           1.400000
......
```

所以最优方案是：用第 4 种方式下料 100 根，第 6 种方式下料 50 根. 最少剩余料头 10 m（最优解不唯一）.

例 3.4 （露天矿生产的车辆安排）露天矿里有若干个爆破生成的石料堆，每堆称为一个铲位，每个铲位已预先根据铁含量将石料分成矿石和岩石. 一般来说，平均铁含量不低于 25%的为矿石，否则为岩石. 每个铲位的矿石、岩石数量，以及矿石的平均铁含量（称为品位）都是已知的. 每个铲位至多能安置一台电铲，电铲的平均装车时间为 5 min. 卸货地点（以下简称卸点）有卸矿石的矿石漏、2 个铁路倒装场（以下简称倒装场）和卸岩石的岩石漏、岩场等，每个卸点都有各自的产量要求. 从保护国家资源的角度及矿山的经济效益考虑，应该尽量把矿石按矿石卸点需要的铁含量（假设要求都为 29.5%±1%，称为品位限制）搭配起来送到卸点，搭配的量在一个班次（8 h）内满足品位限制即可. 从长远看，卸点可以移动，但一个班次内不变. 卡车的平均卸车时间为 3 min.

所用卡车载重量为 154 t，平均时速 28 km/h. 卡车的耗油量很大，每个班次每台车消耗近 1 t 柴油. 发动机点火时需要消耗相当多的电瓶能量，故一个班次中只在开始工作时点火一次. 卡车在等待时所耗费的能量也是相当可观的，原则上在安排时不应发生卡车等待的情况. 电铲和卸点都不能同时为两辆及两辆以上卡车服务. 卡车每次都是满载运输.

每个铲位到每个卸点的道路都是专用的宽 6 m 的双向车道，不会出现堵车现象，每段道路的里程都是已知的.

一个班次的生产计划应该包含以下内容：出动几台电铲，分别在哪些铲位上；出动几辆卡车，分别在哪些路线上各运输多少次. 一个合格的计划要在卡车不等待条件下满足产量和质量（品位）要求，而一个好的计划还应该考虑下面两条原则之一：

（1） 总运量（t · km）最小，同时出动最少的卡车，从而运输成本最小.

（2） 利用现有车辆运输，获得最大的产量（岩石产量优先；在产量相同的情况下，取总运量最小的解）.

请你分别就两条原则建立数学模型，并给出一个班次生产计划的快速算法. 针对下面的实例，给出具体的生产计划、相应的总运量及岩石和矿石产量.

某露天矿有铲位 10 个，卸点 5 个，现有铲车 7 台，卡车 20 辆. 各卸点一个班次的产量要求：矿石漏 1.2×10^4 t、倒装场 I 1.3×10^4 t、倒装场 II 1.3×10^4 t、岩石漏 1.9×10^4 t、岩场 1.3×10^4 t.

铲位和卸点位置的二维示意图如图 3.1 所示，各铲位和各卸点之间的距离（km）见表 3.4.

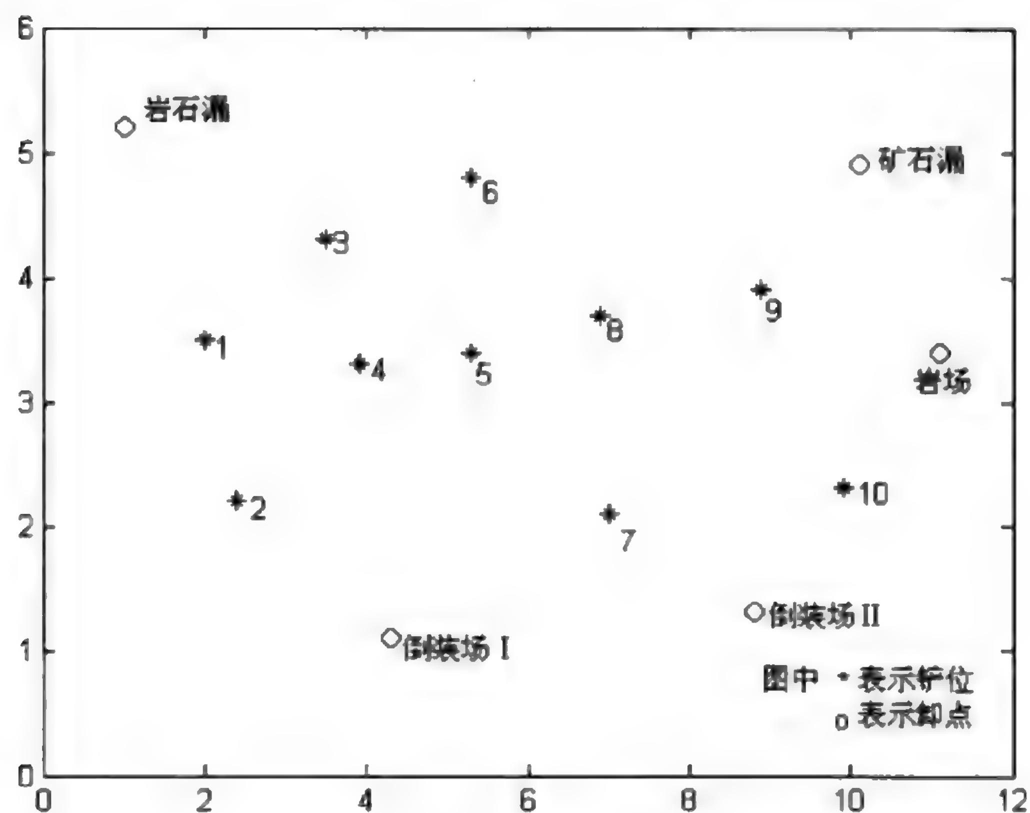


图 3.1　各个铲位和卸点位置的示意图

表 3.4　各铲位和各卸点之间的距离表　　单位:km

	铲位 1	铲位 2	铲位 3	铲位 4	铲位 5	铲位 6	铲位 7	铲位 8	铲位 9	铲位 10
矿石漏	3.26	3.19	4.21	4.00	2.95	2.74	2.46	1.90	0.64	1.27
倒装场 I	1.90	0.99	1.90	1.13	1.27	2.25	1.48	2.0	3.09	3.51
岩场	3.89	3.61	3.61	4.56	3.51	3.65	2.46	2.46	1.06	0.57
岩石漏	0.64	1.76	1.27	1.83	2.74	2.60	4.21	3.72	3.05	6.10
倒装场 II	4.42	3.86	3.72	3.16	2.25	2.81	0.78	1.62	1.27	0.50

各铲位矿石、岩石量和矿石的平均铁含量见表 3.5.

表 3.5　各铲位矿石、岩石量和矿石的平均铁含量表

	铲位 1	铲位 2	铲位 3	铲位 4	铲位 5	铲位 6	铲位 7	铲位 8	铲位 9	铲位 10
矿石量/10^4 t	0.95	1.05	1.00	1.05	1.10	1.25	1.05	1.30	1.35	1.25
岩石量/10^4 t	1.25	1.10	1.35	1.05	1.15	1.35	1.05	1.15	1.35	1.25
铁含量/%	30	28	29	32	31	33	32	31	33	31

分析　本题看起来像运输问题,但与典型的运输问题明显有以下不同:这是运输矿石与岩石两种物资的问题;属于产量大于销量的不平衡运输问题;为了完成品位约束,矿石要搭配运输;产地、销地均有单位时间的流量限制;运输车辆只有一种,每次满载运输,154 t/车次;铲位数多于铲车数意味着要最优的选择不多于 7 个产地作为最后结果中的产地;最后求出各条路线上的派出车辆数及安排.

因而我们可以先求出产位、卸点、每条线路上的运输量.然后求出各条路线上的派出车辆数及安排.

模型假设

(1) 卡车在一个班次中不应发生等待或熄火后再启动的情况.

(2) 在铲位或卸点处由两条路线以上造成的冲突问题面前,我们认为只要平均时间能完成任务,就认为不冲突.我们不排时地进行讨论.

(3) 空载与重载的速度都是 28 km/h,耗油相差很大.

(4) 卡车可提前退出系统.

(5) 铲车一个班次内不移动铲位.

符号说明

x_{ij}:从 i 铲位到 j 卸点的石料运量(车);

c_{ij}:从 i 铲位到 j 卸点的距离(km);

T_{ij}:从 i 铲位到 j 卸点路线上运行一个周期平均时间(min);

A_{ij}:从 i 铲位到 j 卸点最多能同时运行的卡车数(辆);

B_{ij}:从 i 铲位到 j 卸点路线上一辆车最多可运行的次数(次);

p_i:i 铲位的矿石铁含量,$p=(30,28,29,32,31,33,32,31,33,31)\%$;

q_j:j 卸点任务需求,$q=(1.2,1.3,1.3,1.9,1.3)\times 10000(\mathrm{t})$;

ck_i:i 铲位的铁矿石储量(10^4 t);

cy_i:i 铲位的岩石储量(10^4 t);

f_i:描述 i 铲位是否使用的 0-1 变量,取 1 为使用,0 为关闭.

其中

$$T_{ij}=\frac{i\text{ 到 }j\text{ 距离}\times 2}{\text{平均速度}}+3+5,\qquad A_{ij}=\frac{T_{ij}}{5},\qquad B_{ij}=\frac{8\times 60-(A_{ij}-1)\times 5}{T_{ij}}$$

模型建立 建立整数规划模型如下:

$$\min\sum_{i=1}^{10}\sum_{j=1}^{5}x_{ij}\,c_{ij}$$

$$\text{s. t. } x_{ij}\leqslant A_{ij}\,B_{ij},\quad i=1,2,\cdots,10;j=1,2,\cdots,5$$

$$\sum_{j=1}^{5}x_{ij}\leqslant f_i\times 8\times 60/5,\quad i=1,2,\cdots,10$$

$$\sum_{i=1}^{10}x_{ij}\leqslant 8\times 20,\quad j=1,2,\cdots,5$$

$$x_{i1}+x_{i2}+x_{i5}\leqslant ck_i\times 10000/154,\quad i=1,2,\cdots,10$$

$$x_{i3}+x_{i4}\leqslant cy_i\times 10000/154,\quad i=1,2,\cdots,10$$

$$\sum_{i=1}^{10} x_{ij} \geqslant q_j/154, \quad j = 1,2,\cdots,5$$

$$\sum_{i=1}^{10} x_{ij}(p_i - 30.5) \leqslant 0, \quad j = 1,2,5$$

$$\sum_{i=1}^{10} x_{ij}(p_i - 28.5) \geqslant 0, \quad j = 1,2,5$$

$$\sum_{i=1}^{10} f_i \leqslant 7$$

x_{ij} 为非负整数

f_i 为 0-1 整数

约束从上至下分别是：①道路能力（卡车数）约束；②电铲能力约束；③卸点能力约束；④铲位储量约束；⑤产量任务约束；⑥铁含量约束；⑦电铲数量约束；⑧整数约束.

由于程序源码过长，读者请登陆华中数学建模网 http://www.shumo.cn/大学数学实验栏目下载学习本程序.

运行结果如下（表 3.6～表 3.8）：

表 3.6 各铲位和各卸点之间的距离表 单位：km

	铲位 1	铲位 2	铲位 3	铲位 4	铲位 5	铲位 6	铲位 7	铲位 8	铲位 9	铲位 10
矿漏		13						54		11
倒场 I		42		43						
岩场									70	15
岩漏	81		43							
倒场 II		13	2							70

表 3.7 各铲位和各卸点之间的距离表 单位：km

	铲位 1	铲位 2	铲位 3	铲位 4	铲位 5	铲位 6	铲位 7	铲位 8	铲位 9	铲位 10
矿石漏		0.867						1.862		0.314
倒场 I		1.077		1.162						
岩场									1.892	0.326
岩石漏	1.841		1.229							
倒场 II		0.684	0.1							1.489

表 3.8　各铲位和各卸点之间的距离表　　单位:km

	铲位 1	铲位 2	铲位 3	铲位 4	铲位 5	铲位 6	铲位 7	铲位 8	铲位 9	铲位 10
矿石漏								1(29)		
倒场 I		1(39)		1(37)						
岩场									1(37)	
岩石漏	1(44)		1(35)							
倒场 II										1(47)

因而在本思想下的最优结果就是:铲位 1、2、3、4、8、9、10 处各放置一台电铲;一共使用了 13 辆卡车;总运量为 85628.62 t · km;岩石产量为 32186 t;矿石产量为 38192 t.

例 3.5　(选址问题)某公司计划在 A,B,C 三个区建立销售部,确定了 7 个位置 $M_1,M_2,\cdots,M_7$ 可供选择,并且规定:

(1) 在 A 区,从 M_1,M_2,M_3 三个点中至多选两个;

(2) 在 B 区,从 M_4,M_5 两个点中至少选一个;

(3) 在 C 区,从 M_6,M_7 两个点中至少选一个.

已知:如果选择 $M_1,M_2,\cdots,M_7$ 的分别投资为 200,300,350,250,350,200,400 万元,预计每年分别可获利 50,80,120,70,100,60,120 万元,现在公司可用于投资的资金共有 1200 万元,问应如何建立销售部?

模型建立　设选择 $M_1,M_2,\cdots,M_7$ 的投资分别是 b_i 万元,每年获利 c_i 万元,总资金为 b 万元. 设 0-1 变量 $x_i(i=1,2,\cdots,7)$ 如下:

$$x_i=\begin{cases}1, & \text{选择 } M_i \text{ 投资}\\ 0, & \text{否则}\end{cases}$$

则可建立模型如下:

$$\max f=\sum_{i=1}^{7}c_ix_i$$

$$\text{s. t.}\begin{cases}\sum_{i=1}^{7}b_ix_i\leqslant b\\ x_1+x_2+x_3\leqslant 2\\ x_4+x_5\geqslant 1\\ x_6+x_7\geqslant 1\\ x_i=0 \text{ 或 } 1, \quad i=1,2,\cdots,7\end{cases}$$

模型说明　这是一个 0-1 规划问题,当我们面对的对象只有两种结果时,通常引入 0-1 变量来考虑.

0-1 变量和一个量相乘可以表示是否选择这个量,所以模型中 $\sum_{i=1}^{7}c_ix_i$ 并不是代

表把 7 个量加起来，而是在约束条件下限定了取“1”的 0-1 变量所对应的项加起来，所以程序实际上是寻找满足约束的所有的“0,1”组合，经过比较判断来得到最优.

LINGO 程序

```
model:
title 选址问题;
sets:
place/1..7/:c,b,x;
endsets
data:
c=50 80 120 70 100 60 120;
b=200 300 350 250 350 200 400;
enddata
max=@sum(place:c*x);
@sum(place:b*x)<1200;   !资金约束
!在 A 区,从 M1,M2,M3 三个点中至多选两个;
@sum(place(i)|i#le#3:x(i))<2;
!在 B 区,从 M4,M5 两个点中至少选一个;
@sum(place(i)|i#ge#4#and#i#le#5:x(i))>1;
!在 C 区,从 M6,M7 两个点中至少选一个;
@sum(place(i)|i#eq#6#or#i#eq#7:x(i))>1;
@for(place:@bin(x));
End
```

为了让结果只显示非零解，选择 LINGO|Solution，弹出下面面板(图 3.2)：

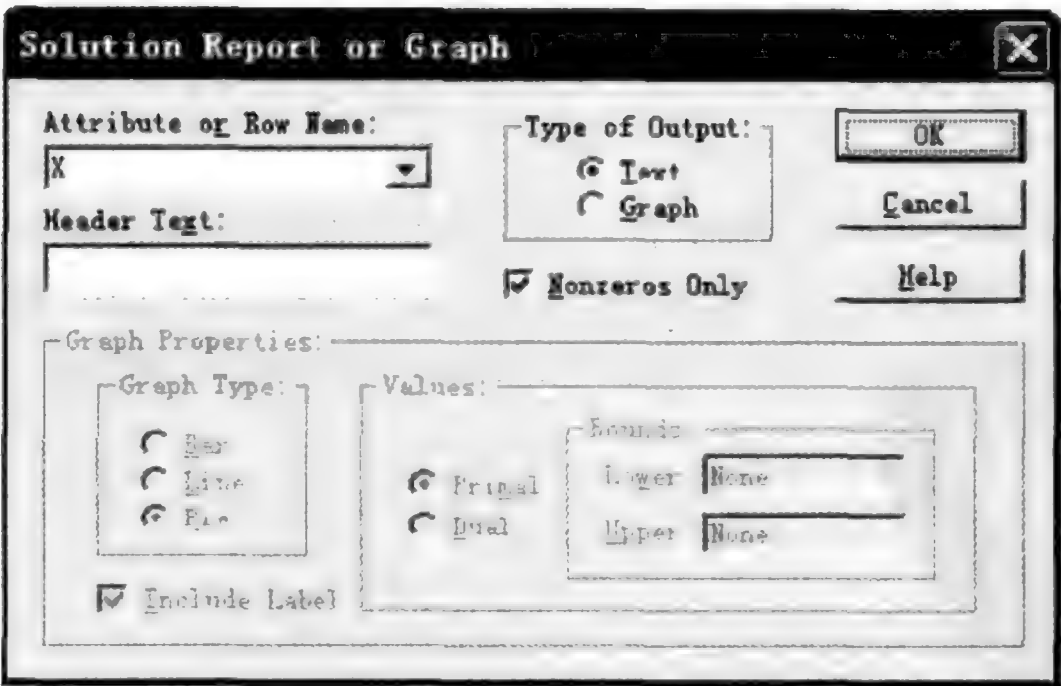


图 3.2　只显示非零解的设置

按上面面板设置确认后，得到报告窗口如下：

```
Global optimal solution found.
   Objective value:                               370.0000
   Extended solver steps:                            0
   Total solver iterations:                          0

   Model Title:选址问题
              Variable           Value        Reduced Cost
                  X(3)        1.000000           -120.0000
                  X(4)        1.000000           -70.00000
                  X(6)        1.000000           -60.00000
                  X(7)        1.000000           -120.0000
```

所以在 M_3, M_4, M_6, M_7 建立销售部，每年盈利 370 万元.

例 3.6 （面试顺序问题）有 4 名同学到一家公司参加三个阶段的面试，公司要求每个同学都必须首先找公司秘书初试，然后到主管部门处复试，最后到经理处参加面试，并且不允许插队，由于 4 名同学的专业背景不同，所以每人在三个阶段的面试时间也不同，见表 3.9，这 4 名同学约定他们全部面试完以后一起离开公司，假定现在时间是早上 8:00，请问他们最早何时能离开公司？

表 3.9　面试时间要求　　　　单位：min

	秘书初试	主管复试	经理面试
同学甲	13	15	20
同学乙	10	20	18
同学丙	20	16	10
同学丁	8	10	15

模型分析与建立　记 t_{ij} 为第 i 名同学参加第 j 阶段面试需要的时间（已知），令 x_{ij} 表示第 i 名同学参加第 j 阶段面试的开始时刻（不妨记早上 8:00 面试开始为 0 时刻）（$i=1,2,3,4; j=1,2,3$），T 为完成全部面试所花费的最少时间.

（1）优化目标.

$$\min T=\{\max_i\{x_{i3}+t_{i3}\}\}$$

（2）约束条件.

个人时间先后次序约束：

$$x_{ij}+t_{ij}\leqslant x_{i,j+1},\quad i=1,2,3,4; j=1,2$$

同阶段不同同学时间不相容（同阶段靠前同学的完成时间小于靠后同学的开始时间）：

$x_{ij}+t_{ij}-x_{kj}\leqslant My_{ik}$

$x_{kj}+t_{kj}-x_{ij}\leqslant M(1-y_{ik})$， M 取大于 T 的常数

$y_{ik}=0$ or 1（取 1 表示 k 排在 i 的前面）

$i,k=1,2,3,4;j=1,2,3$

不妨给定 i,k 的大小关系：$i<k$，可将目标改为如下线性优化目标：

$$\min T$$

$$\text{s.t.}\quad T\geqslant x_{13}+t_{13}$$

$$T\geqslant x_{23}+t_{23}$$

$$T\geqslant x_{33}+t_{33}$$

$$T\geqslant x_{43}+t_{43}$$

LINGO 程序

```
model:
title:面试问题;
sets:
student/1..4/:;
office/1..3/:;
link1(student,office):x,t;
link2(student,student)|&1#lt#&2:y;
endsets
data:
t=13 15 20
   10 20 18
   20 16 10
    8 10 15;
enddata
min=time;
!time大于每名同学最后面试完毕时间;
@for(student(i):time>x(i,3)+t(i,3););
!面试先后次序约束;
@for(student(i):
    @for(office(j)|j#lt#3:x(i,j)+t(i,j)<x(i,j+1)););
!每个阶段只能面试一个同学,y(i,k)=1表示第k名同学排在第i名同学前面;取
 M=1000;
@for(student(i):
     @for(office(j):
           @for(student(k)|k#gt#i:
                 x(i,j)+t(i,j)-x(k,j)<1000*y(i,k))));
```

```
@for(student(i):
    @for(office(j):
        @for(student(k)|k#gt#i:
            x(k,j)+t(k,j)-x(i,j)<1000*(1-y(i,k)))));
!定义 0-1 变量,最后通过 0-1 变量可以查看面试顺序;
@for(link2:@bin(y));
End
```

运行结果如下:

```
Model Title::面试问题
        Variable          Value        Reduced Cost
           TIME        84.00000          0.000000
        ........................(省略)
        Y(1,2)        0.000000         -1000.000
        Y(1,3)        0.000000          0.000000
        Y(1,4)        1.000000          1000.000
        Y(2,3)        0.000000         -1000.000
        Y(2,4)        1.000000          0.000000
        Y(3,4)        1.000000          0.000000
```

所以面试完成至少需要 84 min,面试顺序为 4-1-2-3(丁-甲-乙-丙).

3.2 非线性规划

非线性规划模型可以更具体地表示为如下形式:

$$\begin{aligned} &\min z = f(x) \\ &\text{s.t.}\quad h_i(x)=0,\quad i=1,2,\cdots,l \\ &\qquad\ \ g_j(x)\leqslant 0,\quad j=1,2,\cdots,m \\ &\qquad\ \ x\in \mathbf{R}^n \end{aligned}$$

如果只有等约束 h_i,则可以用拉格朗日乘数法构造拉格朗日函数:

$$L(x,\mu)=f(x)+\sum_{i=1}^{m}\mu_i h_i(x)\quad (\mu_i \text{ 为参数})$$

然后求解非线性方程组$\begin{cases}\dfrac{\partial L}{\partial x_i}=0,\\ \dfrac{\partial L}{\partial \mu_i}=0\end{cases}$即可.

对上述模型,通过讨论 x 的可行方向与下降方向,可得到如下的 KKT (Karush-Kuhn-Tucker) 条件(具体可微等条件略):局部最优解 x 满足(∇为梯度记号)

$$\begin{cases}\nabla f(x)+\sum_{i=1}^{m}\mu_i \nabla h_i(x)+\sum_{j=1}^{l}\lambda_i \nabla g_j(x)=0 \\ \lambda_j g_j(x)=0\end{cases} \quad (\mu_i,\lambda_i \geqslant 0)$$

对等约束模型构造拉格朗日函数：

$$L(x,\mu,\lambda) = f(x) + \sum_{i=1}^{m}\mu_i h_i(x) + \sum_{j=1}^{l}\lambda_j g_j(x)$$

KKT 条件中的公式刚好就是函数 L 对 x 的导数(梯度) 等于 0. (μ,λ) 通常称为拉格朗日乘子.

3.2.1 二次规划

目标函数为二次函数，约束为线性函数的优化问题称为二次规划. 二次规划模型的一般形式：

$$\min f(x)=\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x}+\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$$

$$\text{s.t.} \quad \boldsymbol{A}\boldsymbol{x}\leqslant \boldsymbol{b}$$

$\boldsymbol{H}\in \mathbf{R}^{n\times n}$为对称矩阵. 特别地，当 $\boldsymbol{H}$ 正定时，模型称为凸二次规划. 凸二次规划局部最优解(KKT 点)就是全局最优解.

对等约束的凸二次规划，构造拉格朗日函数：

$$L(x,\lambda)=\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x}+\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}+\boldsymbol{\lambda}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})$$

求导得如下方程组：

$$\begin{cases}\boldsymbol{H}\boldsymbol{x}+\boldsymbol{c}+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\lambda}=\boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}=\boldsymbol{0}\end{cases}$$

解方程组即得最优解.

有效集方法:对于存在不等式约束的二次规划，将等约束(对应有效集)与不等约束分开，将非有效约束去掉，通过解一系列等式约束的二次规划来实现不等式约束的优化.

例 3.7 (投资组合问题)某三支股票 12 年的价格见表 3.10.

表 3.10 1998～2008 年股票价格表

年份	股票 A	股票 B	股票 C	股票指数
1998	1.300	1.225	1.149	1.258 997
1999	1.103	1.290	1.260	1.197 526
2000	1.216	1.216	1.419	1.364 361
2001	0.954	0.728	0.922	0.919 287
2002	0.929	1.144	1.169	1.057 080

续表

年份	股票 A	股票 B	股票 C	股票指数
2003	1.056	1.107	0.965	1.055 012
2004	1.038	1.321	1.133	1.187 925
2005	1.089	1.305	1.732	1.317 130
2006	1.090	1.195	1.021	1.240 164
2007	1.083	1.390	1.131	1.183 675
2008	1.035	0.928	1.006	0.990 108
2009	1.176	1.715	1.908	1.526 236

解决如下问题:如果在 2010 年有一笔资金投资这三种股票,并期望年收益率至少达到 15%,那么应当如何投资?分析投资组合与回报率以及风险的关系.

投资股票的收益是不确定的,因而是一个随机变量,我们可以用期望值来表示.风险可以用方差来衡量,方差越大,风险越大.期望与协方差可由表 3.10 求得.

符号说明

R_1,R_2,R_3:A,B,C 三种股票的收益率,是随机变量;

x_1,x_2,x_3:A,B,C 三种股票的投资比例.

模型建立

目标:
$$\min D(x_1R_1+x_2R_2+x_3R_3)$$

即
$$\min\sum_{j=1}^{3}\sum_{i=1}^{3}x_ix_j\mathrm{Cov}(R_i,R_j)$$

约束:
$$x_1E(R_1)+x_2E(R_2)+x_3E(R_3)\geqslant 0.15$$
$$x_1+x_2+x_3=1$$
$$x_1,x_2,x_3\geqslant 0$$

这是一个二次规划问题.

LINGO 程序

```
model:
title 投资组合模型;
sets:
   year/1..12/;
   stocks/ a,  b,  c/: mean,x;
   link(year,stocks): r;
   stst(stocks,stocks): cov;
endsets
data:
   target=1.15;
```

```
!r是原始数据;
  r=
  1.300    1.225    1.149
  1.103    1.290    1.260
  1.216    1.216    1.419
  0.954    0.728    0.922
  0.929    1.144    1.169
  1.056    1.107    0.965
  1.038    1.321    1.133
  1.089    1.305    1.732
  1.090    1.195    1.021
  1.083    1.390    1.131
  1.035    0.928    1.006
  1.176    1.715    1.908;
enddata
calc:  !计算均值向量 mean 与协方差矩阵 cov;
@for(stocks(i): mean(i)=
   @sum(year(j): r(j,i)) / @size(year));
@for(stst(i,j): cov(i,j)=@sum(year(k):
     (r(k,i)-mean(i))*(r(k,j)-mean(j)))/(@size(year)-1));
endcalc
[obj] min=@sum(stst(i,j): cov(i,j)*x(i)*x(j));
[one] @sum(stocks: x)=1;
[two] @sum(stocks: mean*x)>=target;
end
```

运行结果如下：A 占 53%，B 占 36%，C 占 11%，风险（方差）为 0.0224138，即标准差为 0.1497123.

在数据段，修改赋值语句：

```
TARGET=?;
```

可以输入不同的回报率来观察投资组合以及相应的风险.

3.2.2 一般非线性规划

例 3.8 （供应与选址）建筑工地的位置（用平面坐标 a,b 表示，距离单位：km）及水泥日用量 $d(t)$ 由表 3.11 给出. 有两个临时料场位于 $P(5,1)$，$Q(2,7)$，日储量各有 20 t. 从 A，B 两料场分别向各工地运送多少吨水泥，使总的吨公里数最小. 两个新的料场应建在何处，节省的吨公里数有多大？

表 3.11　建筑工地的位置及水泥日用量表

	1	2	3	4	5	6
a/km	1.25	8.75	0.5	3.75	3	7.25
b/km	1.25	0.75	4.75	5	6.5	7.75
d/t	3	5	4	7	6	11

分析　和运输问题类似，约束涉及供应与需求，是线性的，目标是运量与距离的乘积. 两问的模型都是一样的，第一问知道料场位置，距离是已知的，建立的模型为线性规划模型，第二问不知道料场的位置，距离是未知的，建立的模型为非线性规划模型.

符号说明

(a_j,b_j)，d_j：工地的位置及需求量，$j=1,2,\cdots,6$；

(x_i,y_i)，e_i：料场的位置及日储量，$i=1,2$；

c_{ij}：i 料场向 j 工地的运送量.

模型建立

$$\min f=\sum_{i=1}^{2}\sum_{j=1}^{6}c_{ij}\sqrt{(x_i-a_j)^2+(x_i-b_j)^2}$$

$$\text{s.t.}\quad \sum_{j=1}^{6}c_{ij}\leqslant e_i,\quad i=1,2$$

$$\sum_{i=1}^{2}c_{ij}=d_j,\quad j=1,2,\cdots,6$$

$$c_{ij}\geqslant 0$$

LINGO 程序(第二问)

```
model:
title location problem;
sets:
     demand/1..6/:a,b,d;
     supply/1..2/:x,y,e;
     link(supply,demand):c;
endsets
data:
a=1.25,8.75,0.5,3.75,3,7.25;
b=1.25,0.75,4.75,5,6.5,7.75;
d=3,5,4,7,6,11;e=20,20;
enddata
!初始段:对集合属性定义初值(迭代算法的迭代初值);
```

```
init:
!初始点;
x,y=5,1,2,7;
endinit
min=@sum(link(i,j):c(i,j)*((x(i)-a(j))^2+(y(i)-b(j))^2)^(1/2));
@for(demand(j):@sum(supply(i):c(i,j))=d(j););
@for(supply(i):@sum(demand(j):c(i,j))<=e(i););
@for(supply:@bnd(0.5,X,8.75);@bnd(0.75,Y,7.75););
END
```

直接运行可得到局部最优解，如果调用全局最优求解程序，耗时比较长，这也是求解非线性规划问题的特点. 在程序中，如果将初始段的赋值放到数据段中，即是第一问的求解程序.

例 3.9 （下料问题 2）原料钢管每根长 19 m，现有 4 种需求：4 m 50 根，5 m 10 根，6 m 20 根，8 m 15 根，由于采用不同切割模式太多，会增加生产和管理成本，规定切割模式不能超过 3 种，如何下料最节省？

预处理

```
model:
title 搜索合理的下料方式;
sets:
dm:;
long(dm,dm,dm,dm):;  !有三种需求长度,定义三维数组;
endsets
data:
dm=1..5;
@text('d:\rxl.txt')=@write(3*'','下料方式',9*'','余料长度',@newline(1));
@text('d:\rxl.txt')=@write(16*'-',4*'',8*'-',@newline(1));
@text('d:\rxl.txt')=@writefor(long(i,j,k,m)|
                    (19-8*(i-1)-6*(j-1)-5*(k-1)-4*(m-1))#ge#0
               #and#(19-8*(i-1)-6*(j-1)-5*(k-1)-4*(m-1))#lt#a
         !输出下料方式到文本文件 rxl.txt;
         :i-1,4*'',j-1,4*'',k-1,4*'',m-1,8*'',
19-8*(i-1)-6*(j-1)-5*(k-1)-4*(m-1),@newline(2));
!输出计算段计数过的下料方式总数;
@text('d:rxl.txt')=@write('下料方式总数为:',2*'',n,'种');
Enddata
calc:
a=@smin(8,6,5,4);
b=@floor(19/a)+1;  !用于确定 dm;
```

```
n=0;
@for(long(i,j,k,m)|
                        (19-8*(i-1)-6*(j-1)-5*(k-1)-4*(m-1))#ge#0
                #and#(19-8*(i-1)-6*(j-1)-5*(k-1)-4*(m-1))#lt#a
:n=n+1);  !下料方式计数;
endcalc
end
```

运行程序后在 d:rxl.txt 中输出 16 种下料模式及余料长度，很显然，如果没有下料方式的限制，就需要设置 16 个变量，如果需求种数更多，就会有更多的下料方式，因而先前一般下料问题的做法只适合处理小型问题.

决策变量 由于有下料方式种数的限制，不知道是哪三种下料方式，本着缺什么量就设什么量的原则，设置决策变量如下：x_i 为按第 i 种模式切割的原料钢管根数 $(i=1,2,3)$，$r_{1i}, r_{2i}, r_{3i}, r_{4i}$ 为第 i 种切割模式下，每根原料钢管生产 4 m，5 m，6 m 和 8 m长的钢管的数量.

模型建立

$$\min\ x_1+x_2+x_3$$

$$\text{s.t.}\quad \begin{aligned}
&r_{11}x_1+r_{12}x_2+r_{13}x_3\geqslant 50\\
&r_{21}x_1+r_{22}x_2+r_{23}x_3\geqslant 10\\
&r_{31}x_1+r_{32}x_2+r_{33}x_3\geqslant 20\\
&r_{41}x_1+r_{42}x_2+r_{43}x_3\geqslant 15\\
&16\leqslant 4r_{11}+5r_{21}+6r_{31}+8r_{41}\leqslant 19\\
&16\leqslant 4r_{12}+5r_{22}+6r_{32}+8r_{42}\leqslant 19\\
&16\leqslant 4r_{13}+5r_{23}+6r_{33}+8r_{43}\leqslant 19\\
&x_1\geqslant x_2\geqslant x_3
\end{aligned}$$

$26\leqslant x_1+x_2+x_3\leqslant 31$ %为了缩小可行域便于求解，可取特殊方式确定上下界

$x_i, r_{1i}, r_{2i}, r_{3i}, r_{4i}\ (i=1,2,3)$为整数

LINGO 程序

```
model:
Title 钢管下料-最小化钢管根数的 LINGO 模型;
SETS:
needs/1..4/:length,num;
cuts/1..3/:x;
patterns(needs,cuts):r;
endsets
data:
    length=4 5 6 8;
    num=50 10 20 15;
```

```
    capacity=19;
  enddata
  min=@sum(cuts(I): x(i));
  @for(needs(i):@sum(cuts(j):x(j)*r(i,j))>num(i));  !满足需求约束;
  @for(cuts(j):@sum(needs(i):length(i)*r(i,j))<capacity); !合理切割模式约束;
  @for(cuts(j):@sum(needs(i):length(i)*r(i,j))>capacity-@min(needs(i):
  length(i)));  !合理切割模式约束;
  @sum(cuts(i):x(i))>26;@sum(suts(i):x(i))<31;  !人为增加约束;
  @for(cuts(i)|i#LT#@size(cuts):x(i)>x(i+1));  !人为增加约束;
  @for(cuts(j):@gin(x(j)));
  @for(patterns(i,j):@gin(r(i,j)));
  end
```

求解结果

模式 1:每根原料钢管切割成 3 根 4 m 和 1 根 6 m 钢管,共 10 根;

模式 2:每根原料钢管切割成 2 根 4 m、1 根 5 m 和 1 根 6 m 钢管,共 10 根;

模式 3:每根原料钢管切割成 2 根 8 m 钢管,共 8 根.

用去原料钢管总根数为 28 根.

3.3 多目标规划

3.3.1 基本理论

线性规划只研究在满足一定条件下,单一目标函数取得最优解,而在企业管理中,经常遇到多目标决策问题,例如,拟订生产计划时,不仅考虑总产值,同时要考虑利润、产品质量和设备利用率等. 这些指标之间的重要程度(即优先顺序)也不相同,有些目标之间往往相互发生矛盾.

线性规划致力于某个目标函数的最优解,这个最优解若是超过了实际的需要,很可能是以过分地消耗了约束条件中的某些资源作为代价. 线性规划把各个约束条件的重要性都不分主次地等同看待,这也不符合实际情况.

求解线性规划问题,首先要求约束条件必须相容,如果约束条件中,由于人力、设备等资源条件的限制,使约束条件之间出现了矛盾,就得不到问题的可行解,但生产还得继续进行,这将给人们进一步应用线性规划方法带来困难.

为了弥补线性规划问题的局限性,解决有限资源和计划指标之间的矛盾,在线性规划基础上,建立多目标规划方法,从而使一些线性规划无法解决的问题得到满意的解答.

例 3.10 重新考虑例 3.5 选址问题.

分析 当希望得到最大收益的时候,也希望投资是最少的.

建立模型

$$\max f = \sum_{i=1}^{7} c_i x_i$$

$$\min g = \sum_{i=1}^{7} b_i x_i$$

$$\text{s.t.}\begin{cases}\sum_{i=1}^{7} b_i x_i \leqslant b \\ x_1 + x_2 + x_3 \leqslant 2 \\ x_4 + x_5 \geqslant 1 \\ x_6 + x_7 \geqslant 1 \\ x_i = 0 \text{ 或 } 1, \quad i = 1,2,\cdots,7\end{cases}$$

例 3.11 用直径为 1(单位长)的圆木制成截面为矩形的梁，为使重量最轻，而强度最大，问截面的高与宽应取何尺寸?

设高 x_1，宽 x_2.

$$\min x_1 x_2$$

$$\max \frac{1}{6} x_1 x_2^2$$

$$\text{s.t.} \quad x_1^2 + x_2^2 = 1$$

$$x_1, x_2 \geqslant 0$$

例 3.12 重新考虑例 3.7.

分析 希望风险最小而获利最大.

建立模型

$$\min D(x_1 R_1 + x_2 R_2 + x_3 R_3)$$

$$\max \{x_1 E(R_1) + x_2 E(R_2) + x_3 E(R_3)\}$$

$$\text{s.t.} \quad x_1 E(R_1) + x_2 E(R_2) + x_3 E(R_3) \geqslant 0.15$$

$$x_1 + x_2 + x_3 = 1$$

$$x_1, x_2, x_3 \geqslant 0$$

多目标决策问题有许多共同的特点，其中最显著的是：目标的不可共度性和目标间的矛盾性. 因此不能简单地将多个目标归并为单个目标，并使用单目标决策问题的方法去求解多目标问题.

一般确定性多目标问题的数学模型为

$$\min_{x \in \mathbf{R}^n} f(x) = (f_1(x), f_2(x), \cdots, f_p(x))$$

$$\text{s.t.}\begin{cases} g_j(x) \geqslant 0, & j = 1,2,\cdots,m \\ h_k(x) = 0, & k = 1,2,\cdots,l \end{cases}$$

记可行域为 D.

绝对最优解 x^*：$\forall x \in D, f_i(x) \geqslant f_i(x^*)$. 通常是不存在的，如图 3.5 所示平行

四边形区域 x 取最小值，y 取最小值不可能同时在某个点上达到.

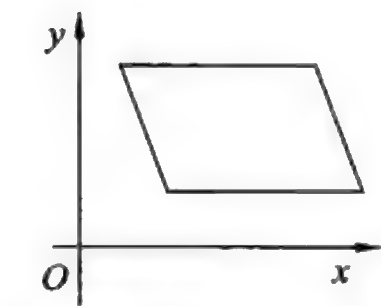


图 3.3　平行四边形区域

有效解(pareto 解)x^*：不存在 $x\in D$，使得 $f_i(x)\leqslant f_i(x^*)$，$i=1,2,\cdots,p$，存在某个 i_0，$f_{i_0}(x)<f_{i_0}(x^*)$. 有效解通常是很多的.

弱有效解 x^*：不存在 $x\in D$，使得 $f_i(x)<f_i(x^*)$，$i=1,2,\cdots,p$. 有点目标达到最优.

5 个基本要素

决策变量：$\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$.

目标函数：$f(x)=(f_1(x),f_2(x),\cdots,f_p(x))$.

可行解集：$X=\{x\in\mathbf{R}^n\mid g_j(x)\geqslant 0,j=1,2,\cdots,m;h_k(x)=0,k=1,2,\cdots,l\}$.

偏好关系：在像集 $f(X)$ 上有某个二元关系$\prec$(称为偏好序)反映决策者的偏好.

解的定义：在已知的偏好关系下定义 f 在 X 上的最好解.

3.3.2　多目标规划的常用解法

思想：转化为单目标问题.

评价函数法　如果能够根据决策者提供的偏好信息构造一个实函数 $u[f(x)]$(称为效用函数)，使得求满意解等价于求以该实函数为新目标函数的单目标规划问题的最优解，则可用已有算法求解.

评价函数法的基本思想是：借助于几何或应用中的直观效果. 构造所谓的评价函数 $u[f(x)]$. 从而将多目标优化问题转化为单目标优化问题. 然后利用单目标优化问题的求解方法求出最优解. 并把这种最优解当作多目标优化问题的最优解. 这里关键的问题是转化后的单目标优化问题的最优解必须是多目标问题的有效解和弱有效解，否则是不能接受的.

由于以下两个原因限制了该方法的实际应用：①许多场合下，决策者提供的偏好信息不足于确定效用函数；②构造实际问题的效用函数是相当困难的.

(1) 线性加权法. 先按目标函数 $f_1(x),f_2(x),\cdots,f_p(x)$的重要程度给一组权数 $\tilde{\omega}_1,\tilde{\omega}_2,\cdots,\tilde{\omega}_p$，满足 $\tilde{\omega}_i\geqslant 0,\sum_{i=1}^{p}\tilde{\omega}_i=1$，然后定义评价函数：$u[f(x)]=\sum_{i=1}^{p}\tilde{\omega}_i f_i(x)$，最后求解非线性规划问题：$\min_{x\in D}u[f(x)]$.

如果每个函数的属性独立于其他函数的属性，则可以用这个方法.

(2) 变权加权法. 在线性加权方法中，一旦多属性效用函数确定了，则其权值就确定了，而不依赖于各属性的效用，有些时候权值是随着其相应属性效用的变化而变化的，此时用变权加权形式：

$$\min_{x\in D} u[f(x)] = \min_{x\in D}\sum_{i=1}^{p}\tilde{\omega}_i[f_i(x)]f_i(x)$$

(3) 指数加权法. 如果各个属性是串行结构，即只要有一个效用为 0，则总体效用为 0，可采用指数加权形式：

$$\min_{x\in D} u[f(x)] = \min_{x\in D}\prod_{i=1}^{p}[f_i(x)]^{\tilde{\omega}_i}$$

(4) 极小极大(min-max) 法. 在决策时，有时候采取保守策略是稳妥的. 即在最坏的情况下，寻求最好的结果. 即求解非线性规划问题：

$$\min_{x\in D} u[f(x)] = \min_{x\in D}\{\max_{1\leqslant i\leqslant p} f_i(x)\}$$

此非线性规划问题目标不可微，不能直接用于基于梯度的算法，可等价转化为

$$\begin{aligned}&\min_{x,t}\ t\\ &\text{s. t.}\quad f_i(x)\leqslant t,\ i=1,2,\cdots,p\\ &\qquad\quad x\in D\end{aligned}$$

(5) 理想点法. 先求解单目标的最优值确定理想点：

$$f_i^* = \min_{x\in D} f_i(x),\quad i=1,2,\cdots,p$$

再找距离理想点最近的 x 点作为最优解：

$$\min_{x\in D} u[f(x)] = \min_{x\in D}\sqrt{\sum_{i=1}^{p}[f_i(x)-f_i^*]^2}$$

(6) 加权偏差函数法.

① 给定理想点：$(\overline{f}_1,\overline{f}_2,\cdots,\overline{f}_p)$，求非线性规划问题：

$$\min_{x\in D} u[f(x)] = \min_{x\in D}\left\{\sum_{i=1}^{p}\tilde{\omega}_i[f_i(x)-\overline{f}_i]^k\right\}^{\frac{1}{k}},\quad \tilde{\omega}_i\geqslant 0,\overline{f}_i\leqslant f_i^*$$

② 采用几何平均函数作为评价函数：

$$\min_{x\in D} u[f(x)] = \min_{x\in D}\prod_{i=1}^{p}[f_i(x)-\overline{f}_i]^{\tilde{\omega}_i},\quad \tilde{\omega}_i>0,\overline{f}_i<f_i^*$$

③ 采用极大模函数：

$$\min_{x\in D} u[f(x)] = \min_{x\in D}\max_{1\leqslant i\leqslant p}\tilde{\omega}_i[f_i(x)-\overline{f}_i]$$

(7) 费效比函数.

$$\min_{x\in D} u[f(x)] = \min_{x\in D}\prod_{i=1}^{k}f_i(x)\Big/\prod_{i=k+1}^{p}f_i(x)$$

特别地，如果目标为 $\min f_1(x)$，$\max f_2(x)$，则可求解 $\max\dfrac{f_2(x)}{f_1(x)}$，若 f_1 为投资，f_2 为收益，则计算的结果为单位投资的总收入最大.

(8) 功效系数函数. 对不同的性质的目标函数统一量纲，再构造效用函数：

$$f_{i,\min} = \min_{x\in D} f_i(x),\quad i=1,2,\cdots,p,\qquad f_{i,\max} = \max_{x\in D} f_i(x),\quad i=1,2,\cdots,p$$

例如,构造功效系数函数:

$$d_i(x)=\frac{f_{i,\max}-f_i(x)}{f_{i,\max}-f_{i,\min}}\in[0,1]$$

然后求解规划问题:

$$\max_{x\in D}\sum_{i=1}^{p}d_i(x)\quad 或\quad \max_{x\in D}\prod_{i=1}^{p}d_i(x).$$

还可以对功效系数函数进行加权构造效用函数,如

$$u(x)=\left(\prod_{i=1}^{p}\{d_i[f_i(x)]\}^{\varpi_i}\right)^{1/\sum_{i=1}^{p}\varpi_i}$$

(9) 参考目标法.约束法:在多个目标中选定一个主要目标,而对其他目标设定一个期望值,在要求结果不比期望值坏的情况下,求主要目标的最优值.例如,第一个目标设为主要目标的话,设定期望值$(f_2^0,f_3^0,\cdots,f_p^0)$有

$$\begin{aligned}&\min f_1(x)\\ &\text{s.t.}\quad f_i(x)\leqslant f_i^0,\quad i=2,3,\cdots,p\\ &\qquad\ x\in D\end{aligned}$$

优点:适当的选择目标函数值,可以方便地得到原问题的每一个有效解,而且重点保证了重要目标的效益,同时又照顾了其他目标,而且在当前点的 Kuhn-Tucker 乘子可用来确定置换率,帮助决策者寻找更合意的方案.

(10) 分层序列法.把多个目标按照重要程度进行排序,先求第一个目标的最优解,在达到此目标的条件下求第二个目标的最优解,以此类推直到求得最后一个目标的最优解.

缺点:当先前的目标只有唯一的最优解时,后面的求解失去了意义.

改进 - 宽容分层序列法:给前面的最优值设置一定的宽容值,和最优值相差宽容值之内的都是可以接受的.

(11) 逼近理想解法.在多目标问题的决策过程中,决策人总是希望找到所有属性指标都为最优的解,即希望尽可能地远离各属性指标都最劣的解.所有属性指标都处于最优的解称为正理想解,所有属性指标都处于最劣的解称为负理想解,从几何上看,若一个方案在某种测试下,最靠近理想解,而又最远离负理想解,则该方案就认为是决策问题的最优解.

正负理想解:

$$f_i^+=\min_{x\in D}f_i(x),\qquad f_i^-=\max_{x\in D}f_i(x),\quad i=1,2,\cdots,p$$

计算距离,不妨取为欧式距离:

$$d^+(x)=\left\{\sum_{i=1}^{p}[f_i^+-f_i(x)]^2\right\}^{1/2},\quad d^-(x)=\left\{\sum_{i=1}^{p}[f_i(x)-f_i^-]^2\right\}^{1/2}$$

计算测度: $$c(x)=\frac{d^-(x)}{d^-(x)+d^+(x)}$$

求最大测度：

$$\max_{x \in D} c(x)$$

并以此解作为多目标决策问题的最优解.

例 3.13 某企业拟生产 A 和 B 两种产品，其生产投资费用分别为 2100 元/t 和 4800 元/t. A,B 两种产品的利润分别为 3600 元/t 和 6500 元/t. A,B 产品每月的最大生产能力分别为 5 t 和 8 t；市场对这两种产品总量的需求每月不少于 9 t. 试问该企业应该如何安排生产计划，才能既满足市场需求，又节约投资，而且使生产利润达到最大？

该问题是一个线性多目标规划问题. 如果计划决策变量用 x_1 和 x_2 表示，它们分别代表 A,B 产品每月的生产量（单位：t），$f_1(x_1, x_2)$分别表示生产 A,B 两种产品的总投资费用（单位：元），$f_1(x_1, x_2)$分别表示生产 A,B 两种产品获得的总利润（单位：元）. 那么，该多目标规划问题就是：求 x_1 和 x_2，使

$$\min f_1(x_1, x_2) = 2100x_1 + 4800x_2$$

$$\max f_2(x_1, x_2) = 3600x_1 + 6500x_2$$

而且满足：

$$\begin{cases} x_1 \leqslant 5 \\ x_2 \leqslant 8 \\ x_1 + x_2 \geqslant 9 \\ x_1, x_2 \geqslant 0 \end{cases}$$

在本例中，采用偏差加权函数法求解. $k=2$，$f_1=30000$，$f_2=45000$，f_1，f_2 为单目标问题的最优解. $\overline{w_1}=30000$，$\overline{w_2}=450000$.

求解程序如下：

```
model:
sets:
yuefen/1,2/:x,f;
endsets
min=30000*f(1)+45000*f(2);  !30000 45000 为权重;
f(1)=(2100*x(1)+4800*x(2)-30000)*(2100*x(1)+4800*x(2)-30000);
!30000 为目标值;
f(2)=(3600*x(1)+6500*x(2)-45000)*(3600*x(1)+6500*x(2)-45000);
!450000 为目标值;
x(1)<5;
x(2)<8;
x(1)+x(2)>9;
end
```

运行结果如下：

```
Global optimal solution found.
Objective value:                          0.3352500E+12
Extended solver steps:                              2
Total solver iterations:                          513

                  Variable          Value      Reduced Cost
                      x(1)       4.000000       0.9179959E+12
                      x(2)       5.000000       0.1802692E+13
```

由程序运行结果可知，应该生产 A 产品 5 t，B 产品 4 t.

评论　相对于单目标规划模型，多目标规划模型具有很强的实际意义，其难点在于求解，在求解时应当注意根据目标之间的关系及量纲，选取合适的求解方式.

3.4　目标规划

线性规划有着完整的理论与求解方法，也有很好的应用，但是线性规划也有很大的局限性：

(1) 只能处理单目标问题，而实际中经常碰到多目标问题，而且目标与约束是可以转化的.

(2) 约束都是刚性条件，需要严格满足，而实际情况中，并不是这样严格，而且过于严格可能导致无解，不利于我们做出决策.

(3) 约束或者目标看成同等重要，而实际中各个目标可能有重要性的差别.

(4) 给出的是最优解，而许多实际问题只需要找到满意解即可.

例如，在企业安排生产问题中，既希望利润高，又要消耗低，还要考虑市场上产品的销路等. 当然，这些目标之间往往是相互矛盾的，要追求利润最大，通常消耗便不可能最低. 而且目标具有"柔性"，能否构造这样一个数学模型，其结果既使利润尽量地大，同时使消耗尽量地低，销路尽量地好呢？目标规划(goal programming)便用来处理这样的问题.

3.4.1　目标规划的数学模型

先来看一个引例.

例 3.14　(多目标生产计划问题)某工厂计划用所拥有的三种资源生产代号为 A,B 的两种产品，原材料资源可供量为 90 t，使用专用设备台时最多为 200 台时，劳动力 300 个；生产单位产品 A 需用原材料 2.5 t，设备台时 4 个和劳动力 3 个，产品 B 则需用原料 1.5 t，设备台时 5 个和劳动力 10 个. 扣除成本，每单位产品 A,B 分别可获利 7 万元和 12 万元. 求一个生产计划，使获利最大.

据题设，设 x_1,x_2 依次表示产品 A 和 B 的生产量，容易得到其线性规划基础模型为

$$\max f=7x_1+12x_2$$

$$\text{s.t.}\quad 2.5x_1+1.5x_2\leqslant 90$$

$$4x_1+5x_2\leqslant 200$$

$$3x_1+10x_2\leqslant 300$$

$$x_1,x_2\geqslant 0$$

可求出其绝对最优解为 $x_1=20,x_2=24$，最大利润值为 428 万元.

现增加以下考虑：

(1) 上述结果并未考虑市场信息和资源的可塑性条件，仅仅根据现有生产能力和固定不变的产品价格求得，因而是脱离实际的"理想化"方案. 依据市场调查和生产能力，厂长认为上述利润指标不易达到，决定降低为 420 万元，当然力求超过.

(2) 根据市场调查和预测，产品 B 开始出现滞销现象，随着市场需求的改变，预测两种产品的需求量比例大致为 1∶1，而目前的产品比例失调，有待调整.

(3) 根据原材料市场信息，这种原材料的市场价格下跌，而所生产的产品价格基本稳定，故决策者希望尽量将原材料转化为产品，即希望原料要全部用掉. 但按原生产计划看，原材料将有剩余(4 t). 因此，尽可能将原材料全部转化为利润成为一个重要的生产规划指标.

为了叙述方便，先来考虑单利润指标情况. 实现利润 420 万元是决策者的希望，但在计划具体实施后，由于各方面因素的制约，完全有可能达不到，也完全可能超过该指标，换句话说，可能实现的利润指标和规定的利润指标完全可能不一致而产生某一差距. 我们称这个差距为偏差变量，记为 d，规定 $d\geqslant 0$. 从决策者的心理和要求来分析，使之绝对满意可以做不到，但他总希望将来得到的实际利润与规定的指标值之间偏差量愈小愈好，这就"等价地"表出了他希望利润值达到 420 万元的目标. 当然，他所希望的是未达规定指标的实际值与规定值的偏差量越小越好.

由于可能达到并超过，也可能达不到，我们引入下述符号：d^+ 表示超出指标的偏差变量，称为正偏差变量，d^- 表示未达指标的偏差变量，称为负偏差变量. 自然规定 $d^+,d^-\geqslant 0$. 显然，偏差变量 d^+,d^- 的取值有且仅有下述三种情形：

(1) 超额完成指标时，$d^+>0,d^-=0$.

(2) 未能完成指标时，$d^+=0,d^->0$.

(3) 恰好完成指标时，$d^+=0,d^-=0$.

有了偏差变量的概念，上述利润指标就可以比较灵活地进行表示了. 事实上，决策者的目标是利润达到或超过 420. 因此，他所希望的自然是 $d^+>0$. 但实际中完全可能 $d^->0$，这是决策者所不希望出现的，而一但出现 $d^->0$，也希望 d^- 尽可能地小. 因此，决策者最关心的是 d^- 达到最小，故此时的目标函数可表示为 $\min d^-$. 这样，我们把目标函数写成了偏差变量的函数. 注意，例子中原来的目标函数显然不再成

为目标规划的目标函数，由于它在目标规划中只是问题要达到的目标之一，因而也成了一个约束条件. 事实上，作为目标之一的利润值已被限制（约束）在 420 万元，用偏差变量很容易将它表示为 $7x_1+12x_2+d^- - d^+ = 420$. 它自然是约束条件，而且确切地表出了目标利润应为 420 万元这一约束. 事实上，当达不到 420 万元时，由于 $d^+=0$，从而 $d^- = 420-(7x_1+12x_2)>0$，即 $d^+=0, d^->0$，恰好说明利润指标未达要求；当超过 420 万元时，由于 $7x_1+12x_2>420$，$d^-=0$，故 $d^+=(7x_1+12x_2)-420>0$；而当利润恰为 420 万元时，因 $d^- - d^+=0$，故有 $d^-=d^+=0$.

由于这一约束条件是目标规划的目标之一的约束要求，故又称为目标约束，其特点是带有偏差变量的等式约束. 凡非目标约束的约束条件统称为系统约束或刚性约束. 也相应地称目标约束为柔性约束，这主要是因为这种约束比刚性约束灵活.

至此，我们可将上述单利润指标的规划问题写成如下形式：

$$\min d^-$$
$$\text{s. t.}\quad 7x_1+12x_2+d^- - d^+ = 420$$

称这种规划模式为目标规划模式. 其特点主要有两条，其一是目标函数是各目标的偏差变量的函数，其二是约束条件中含有目标约束条件.

有了上面关于单指标目标规划的构模原理，我们来讨论具有三个指标情形的该如何构模.

将上述偏差变量给以下标 1 表示利润指标对应的偏差变量：

$$\min d_1^-$$
$$\text{s. t.}\quad 7x_1+12x_2+d_1^- - d_1^+ = 420$$

现考虑原材料要求的指标. 设 d_2 表示原材料偏差变量，则 d_2^- 表示未用量，d_2^+ 表示超用量，由于希望 90 t 原料全部用完，既不希望有余也不希望超支，故目标函数应为 $\min\{d_2^- + d_2^+\}$. 相应的目标约束完全类似利润指标情形应为 $2.5x_1+1.5x_2+d_2^- - d_2^+ = 90$. 又设 d_3 表示产品比例指标偏差变量. 即以 d_3 表示产品 A 和 B 之产量差距，则由于要求两种产品的产量尽可能达到或接近 1∶1，故有目标函数 $\min\{d_3^- + d_3^+\}$，其对应的约束条件可写为 $x_1-x_2+d_3^- - d_3^+ = 0$.

这是一个新增加的目标约束，当两产品产量一致时，$x_1-x_2=0$，从而 $x_1:x_2=1:1$.

我们的目的是求这三个目标的统一体的最优化方案，即从整体看，我们希望各个指标的偏差总和最小，要把所涉及指标都考虑到，只能按各目标的轻重缓急分级考虑. 事实上，各目标的重要程度是不同的，可以因人，因地，因时而异. 例如，产品的产量问题对有的企业来讲是第一位的，而对别的企业来讲则是第二位的；同一企业此时此地产量第一而彼时彼地可能产量就放在第二位甚至于第三位，等等. 这就需要决策者或决策集团根据各目标的重要程度，科学地予以排队.

我们规定 p_1 表示第一位重要，p_2 表示第二位重要，即满足 $p_1 \gg p_2$. 称 p_k 为优先因子. 例如，第一位重要的目标是要求超额完成利润指标，则赋予它优先因子 p_1，其在整

个问题的目标函数中表为 $p_1d_1^-$，列为第一优先级；其次要目标是要求恰好用完原材料，赋予优先因子 p_2，列为第二优先级表为 $p_2(d_2^-+d_2^+)$，最后是要求产品产量比例达 1:1 的目标，赋予优先因子 p_3，列为第三优先级，在整体目标函数中表为 $p_3(d_3^-+d_3^+)$.

于是整个问题的目标函数表为

$$\min z=p_1d_1^-+p_2(d_2^-+d_2^+)+p_3(d_3^-+d_3^+)$$

从而三个目标的的目标规划模型为

$$\begin{aligned}&\min z=p_1d_1^-+p_2(d_2^-+d_2^+)+p_3(d_3^-+d_3^+)\\ &\text{s.t.}\quad 4x_1+5x_2\leqslant 200\\ &\qquad 3x_1+10x_2\leqslant 300\\ &\qquad 7x_1+12x_2+d_1^--d_1^+=420\\ &\qquad 2.5x_1+1.5x_2+d_2^--d_2^+=90\\ &\qquad x_1-x_2+d_3^--d_3^+=0\\ &\qquad x_1,x_2\geqslant 0,\quad d_i^-,d_i^+\geqslant 0,\quad i=1,2,3\end{aligned}$$

目标规划的一般模型：设 $x_j\ (j=1,2,\cdots,n)$ 是目标规划的决策变量，共有 m 个约束是刚性约束，有 l 个柔性目标约束，其目标规划约束的偏差为 $d_i^-,d_i^+\ (i=1,2,\cdots,l)$，设有 q 个优先级别，分别为 $p_1,p_2,\cdots,p_q$，在同一个优先级别 p_k 中，不同的目标有不同的权重（重要性的区别），分别记为 $\tilde{\omega}_{kj}^+,\tilde{\omega}_{kj}^-$，目标规划模型的一般数学表达式为

$$\begin{aligned}&\min z=\sum_{k=1}^{q}p_k\sum_{j=1}^{l}(\tilde{\omega}_{kj}^-d_j^-+\tilde{\omega}_{kj}^+d_j^+)\\ &\text{s.t.}\quad \sum_{j=1}^{n}a_{ij}x_j\leqslant(=,\geqslant)b_i,\quad i=1,2,\cdots,m\\ &\qquad \sum_{j=1}^{n}c_{ij}x_j+d_i^--d_i^+=g_i,\quad i=1,2,\cdots,l\\ &\qquad x_j\geqslant 0,\quad j=1,2,\cdots,n\\ &\qquad d_i^-,d_i^+\geqslant 0,\quad i=1,2,\cdots,l\end{aligned}$$

确定目标函数的一般原则：

(1) 对于约束都是固定的形式，在刚性约束中添加 $d_i^--d_i^+$ 后都取等号.

(2) 对于目标函数，

① 如果要求恰好达到目标，即要求目标的正负偏差都尽可能的小，则取

$$\min z=f(d_k^-+d_k^+)$$

② 如果要求超过指标值，即要求目标的正偏差不限，而负偏差越小越好，则取

$$\min z=f(d_k^-)$$

③ 如果要求超过指标值，即要求目标的负偏差不限，而正偏差越小越好，则取

$$\min z=f(d_k^+)$$

④ 再根据各目标的优先级赋予相应的优先因子和加权系数.

3.4.2 目标规划模型的求解

对目标规划模型(我们主要讨论线性模型)的求解方法有两种,一种是直接使用单纯形法,结合优先级别 p_k 来判断判别数的情况. 我们主要介绍另一种方法 —— 序贯式算法.

序贯式算法的思想:将目标规划模型按照各目标的优先等级次序,依次分解为若干个单目标的规划问题,按级别高低依次求解,优先级高的最优解求出后,作为该目标偏差的上界添加到低级别问题中作为约束条件,再对低级规划问题求解.

我们可以采用 LINGO 软件实现这个算法. 3.4.1 节的问题 LINGO 程序编写如下:

```
model:
title 目标规划;
sets:
  level/1..3/: p,z,goal;
  variable/1..2/: x;
  h_con_num/1..2/: b;
  s_con_num/1..3/: g,dplus,dminus;
  h_cons(h_con_num,variable): a;
  s_cons(s_xon_num,variable): c;
  obj(level,s_con_num): wplus,wminus;
endsets
data:
  p=  ? ? ?;
  goal =  ? ? 1000;
  b =  200 300;
  g=   420 90 0;
  a =  4 5
      3 10;
  c =  7   12
      2.5 1.5
       1  - 1  ;
  wplus =    0 0 0
             0 1 0
             0 0 1;
  wminus =  1 0 0
             0 1 0
```

```
            0 0 1;
enddata

min=@sum(level: p * z);
@for(level(i):
   z(i)=@sum(s_con_num(j): wplus(i,j)*dplus(j))
        +@sum(s_con_num(j): wminus(i,j)*dminus(j)));
@for(h_con_num(i):
   @sum(variable(j): a(i,j) *  x(j)) <=  b(i));
@for(s_con_num(i):
@sum(variable(j): c(i,j)*x(j))
  +  dminus(i) -  dplus(i) =  g(i);
);
@for(level(i) | i #lt#@size(level):
  @bnd(0,z(i),goal(i));
);
end
```

需要求解三次才能得到最后解.

第一次求解:P(1),P(2),P(3)分别输入1,0,0,Goal(1),Goal(2)输入1000,1000(较大的值即可,让约束不起作用),得到最优偏差为0.

第二次求解:P(1),P(2),P(3)分别输入0,1,0,Goal(1),Goal(2)输入0(为第一次的最优偏差),1000,得到最优偏差为0.

第三次求解:P(1),P(2),P(3)分别输入0,0,1,Goal(1),Goal(2)输入0,0(为第二次的最优偏差),得到最优偏差为1.538462,进而求得最优利润.

评论 目标规划在解决具有柔性约束时具有显著的优点,具有根据实际需求进行求解的特点.在使用时要注意与多目标规划区别.

第4章 图 论

图论起源于著名的柯尼斯堡七桥问题. 在柯尼斯堡的普莱格尔河上有7座桥将河中的岛及岛与河岸连接起来，如图4.1所示，A,B,C,D 表示陆地. 问题是要从这4块陆地中任何一块开始，通过每一座桥正好一次，再回到起点. 然而无数次的尝试都没有成功. 欧拉在1736年解决了这个问题，他用抽象分析法将这个问题化为第一个图论问题，即把每一块陆地用一个点来代替，将每一座桥用连接相应的两个点的一条线来代替，从而相当于得到一个图. 欧拉证明了这个问题没有解，并且推广了这个问题，给出了对于一个给定的图可以某种方式走遍的判定法则. 这项工作使欧拉成为图论(及拓扑学)的创始人.

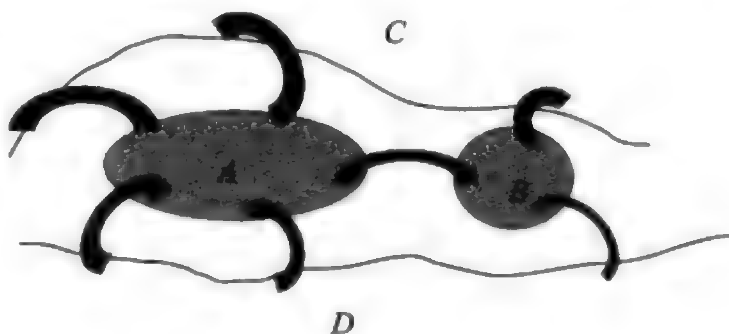


图4.1 柯尼斯堡七桥图

1859年，英国数学家哈密顿发明了一种游戏：用一个规则的实心十二面体，它的20个顶点标出世界著名的20个城市，要求游戏者找一条沿着各边通过每个顶点刚好一次的闭回路，即"绕行世界". 用图论的语言来说，游戏的目的是在十二面体的图中找出一个生成圈. 这个问题后来就称为哈密顿问题. 由于运筹学、计算机科学和编码理论中的很多问题都可以化为哈密顿问题，这一问题引起广泛的注意和研究.

在图论的历史中，还有一个最著名的问题——四色猜想. 这个猜想说，在一个平面或球面上的任何地图能够只用四种颜色来着色，使得没有两个相邻的国家有相同的颜色. 每个国家必须由一个单连通域构成，而两个国家相邻是指它们有一段公共的边界，而不仅仅只有一个公共点. 每个地图可以导出一个图，其中国家都是点，当相应的两个国家相邻时这两个点用一条线来连接. 所以四色猜想是图论中的一个问题. 它对图的着色理论、平面图理论、代数拓扑图论等分支的发展起到推动作用.

在本章中，我们将重点讲解有关图论问题的各种算法.

4.1 图的基本概念

图论中的"图"并不是通常意义上的几何图形或物体的形状图，而是以一种抽象的形式来表达一些确定的事物之间的联系的一个数学系统. 从这个意义上说，有关事物间联系的问题，都可以从图的角度加以分析，这样，图的一些基本方法的运用是

很重要的. 数学建模竞赛中对图的方法的运用也是非常多的.

定义 4.1 一个有序二元组 (V,E) 称为一个图,记为

$$G=(V,E)$$

其中 V 称为图 G 的顶点集,其元素称为顶点;E 称为图 G 的边集,其元素称为边.

定义 4.2 若将图 G 的每一条边 e 都对应一个实数 $F(e)$,则称 $F(e)$ 为该边的权,并称图 G 为赋权图(网络),记为 $G=(V,E,F)$.

定义 4.3 一个 n 阶赋权图 $G=(V,E,F)$ 的权矩阵为 $\mathbf{A}=(a_{ij})_{n\times n}$,其中

$$a_{ij}=\begin{cases}F(v_iv_j), & v_{ij}\in E\\ 0, & i=j\\ +\infty, & v_{ij}\notin E\end{cases}$$

作为阅读此章的基础,图的更多概念请读者参考参考文献《数学方法选讲》(刘承平,2002).

4.2 Dijkstra 算法与 Warshall-Ford 算法

最短路径问题(shortest route problem)是竞赛中常见的一类中等难度的求解问题,如 1999 年国赛的钢管运输问题,甚至有时一些看似跟最短路径问题无关的问题也可以归结为最短路径问题. 但实际问题通常由于结点数非常多,让人感觉困难重重,无法入手. 解这类题时同学们往往不得要领,不少同学想采用穷举法把所有可能的情况全部列出,再找出其中最短的那条路径;或是采用算法中的递归或深度搜索算法寻找. 这两种方法都是费时非常多的,如果城市数目过多则很可能要超时,显然这几种算法对于求最短路径这类最优解问题是不合适的,所以在数模竞赛中求解时运用得非常少,通常也不建议读者使用. 本节简要地和读者一起探讨一下该类问题,分析一下此类问题的一些算法,也使有兴趣的同学对此问题有初步了解.

在现实生活中,经常需要用到两点间的最短距离. 求两点间的最短路及最短距圈离,常用的为 Dijkstra 算法与 Warshall-Ford 算法.

4.2.1 Dijkstra 算法与动态规划

动态规划算法已经成为了许多求解的重要算法,只不过在很多题目中动态规划的表达式比较难写,而恰恰图论中求某两点的最短路问题如果用动态规划算法考虑则可以非常容易地找到那个表达式. 在图论问题中,实现动态规划求解某两点的最短路问题的算法即 Dijkstra 算法.

例 4.1 求图 4.2 中从 P_1 到 P_5 的最短路.

记 $d(Y,X)$ 为城市 Y 与城市 X 之间的直接距离(若这两个城市之间没有道路直接相连,则可以认为直接距离为 $+\infty$),用 $L(X)$ 表示城市 P_1 到城市 X 的最短路线的

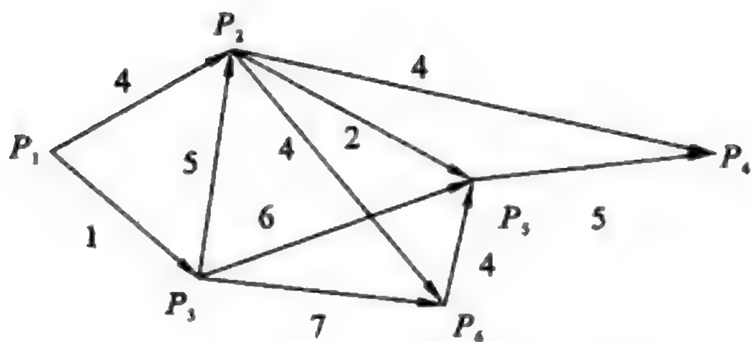


图 4.2　求 P_1 到 P_5 的最短路

路长，则模型可写为

$$\begin{cases} L(P_1)=0 \\ L(X)=\min\limits_{Y\neq X}\{L(Y)+d(Y,X)\}, \quad X\neq P_1 \end{cases}$$

算法适用的条件和范围：

(1) 求图 G 中某一点到其他各点最短路，一般用 Dijkstra 算法.

(2) 有向图和无向图（无向图可以视为 (u,v)，(v,u) 同属于边集 E 的有向图）均适用.

(3) 所有边权非负（任取 $(i,j)\in E$ 都有 $d_{ij}\geqslant 0$）.

利用 Dijkstra 算法我们也可以写出最短路问题的程序代码. 由于程序源码过长，读者请登陆华中数学建模网 http://www.shumo.cn/大学数学实验栏目下载学习.

4.2.2　Warshall-Ford 算法

Warshall-Ford 算法是求任意两顶点间最短距离的算法.

设 $\boldsymbol{A}=(a_{ij})_{n\times n}$ 为赋权图 $G=(V,E,F)$ 的权矩阵，当 $v_iv_j\in E$ 时，$a_{ij}=F(v_iv_j)$，否则取 $a_{ii}=0$，$a_{ij}=+\infty\ (i\neq j)$，$d_{ij}$ 表示从 v_i 到 v_j 点的距离，r_{ij} 表示从 v_i 到 v_j 点的最短路中一个点的编号.

① 赋初值. 对所有 i,j，$d_{ij}=a_{ij}$，$r_{ij}=j$，$k=1$. 转向②.

② 更新 d_{ij}，r_{ij}，对所有 i,j，若 $d_{ik}+d_{kj}<d_{ij}$，则令 $d_{ij}=d_{ik}+d_{kj}$，$r_{ij}=k$，转向③.

③ 终止判断. 若 $d_{ii}<0$，则存在一条含有顶点 v_i 的负回路，终止；或者 $k=n$ 终止；否则令 $k=k+1$，转向②.

其中最短路线可由 r_{ij} 得到.

该算法的适用条件和范围：①任意两点间的最短路径；②可以适用于有负权的情况.

算法较简单，适用范围广，实践中的效果不错，虽然复杂性略高（时间复杂度 $O(n^3)$），仍不失为一个很实用的算法.

下面用一个例子来看看 Floyd 最短路算法的 MATLAB 程序：

例 4.2　求图 4.3 中任意两点间的最短路.

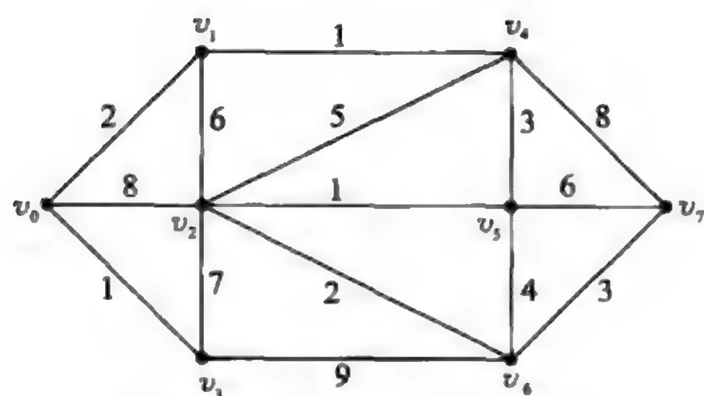


图 4.3　8 个点的连通赋权图

用 Warshall-Floyd 算法，MATLAB 程序代码如下：

```
        %floyd.m
  function[D,R]=floydwarshall(A)
        %采用 floyd 算法计算图中任意两点之间最短路程,可以有负权
        %参数 D 为连通图的权矩阵
        %R 是路线矩阵
  D=A;n=length(D);  %赋初值
  for(i=1:n)for(j=1:n)R(i,j)=j;end;end  %赋路径初值
  for(k=1:n)
      for(i=1:n)
          for(j=1:n)
      if(D(i,k)+D(k,j)<D(i,j))D(i,j)=D(i,k)+D(k,j);
  %更新 dij,说明通过 k 的路程更短
      R(i,j)=k;end;end;end  %更新 rij,需要通过 k
      k  %显示迭代步数
      D  %显示每步迭代后的路长
      R  %显示每步迭代后的路径
      pd=0;for i=1:n%含有负权时
         if(D(i,i)<0)pd=1;break;end;end
  %跳出内层的 for 循环,存在一条含有顶点 vi 的负回路
      if(pd==1)  fprintf('有负回路');break;end
  %存在一条负回路,跳出最外层循环,终止程序
  end  %程序结束
```

在命令窗口输入：

```
  n=8;
  A=[  0    2    8    1    Inf   Inf   Inf   Inf
       2    0    6    Inf  1     Inf   Inf   Inf
       8    6    0    7    5     1     2     Inf
       1    Inf  7    0    Inf   Inf   9     Inf
```

```
      Inf   1    5    Inf  0     3     Inf   8
      Inf   Inf  1    Inf  3     0     4     6
      Inf   Inf  2    9    Inf   4     0     3
       Inf  Inf  Inf  Inf  8     6     3     0];  %MATLAB 中,Inf 表示+∞
   [D,R]=floydwarshall(A)
```

运行结果(部分)如下:

```
   D =
        0     2     7     1     3     6     9    11
        2     0     5     3     1     4     7     9
        7     5     0     7     4     1     2     5
        1     3     7     0     4     7     9    12
        3     1     4     4     0     3     6     8
        6     4     1     7     3     0     3     6
        9     7     2     9     6     3     0     3
       11     9     5    12     8     6     3     0
```

其中 D 矩阵为任意两点间的最短距离.

评论 Dijkstra 算法与 Warshall-Ford 算法作为求解点与点之间最短距离及相应路径的算法,具有简单易懂,运行速度快的特点,在使用时要注意两者的区别,根据实际需要进行算法的选择.

4.3 最小生成树

定义 4.4 连通而无圈的图称为树(tree),常用 T 表示树.

设 $G=(V,E)$ 是一个无向连通赋权图. E 中每条边 (v_i,v_j) 的权为 a_{ij}. 如果 G 的一个子图 G' 是一棵包含 G 所有顶点的树,则称 G' 为 G 的生成树. 或者可以由树的定义认为,任意一个连通的 (p,q) 图 G 适当去掉 $q-p+1$ 条边后,都可以变成树,这棵树称为图 G 的生成树.

关于树的一些结论:

(1) 设 G 是连通图,且边数小于顶点数,则 G 中至少有一个顶点的度为 1.

(2) 设 G 是具有 n 个顶点的无向连通图,G 是树的充分必要条件是:G 有 $n-1$ 条边.

最小生成树问题:设 T 是图 G 的一棵生成树,用 $F(T)$ 表示树 T 中所有边的权数之和,$F(T)$ 称为该生成树 T 的权(费用). 一个连通图 G 的生成树一般不止一棵,在 G 的所有生成树中,权数最小的生成树称为 G 的最小生成树.

网络的最小生成树在实际中有广泛应用. 例如,在设计通信网络时,用图的顶点表示城市,用边 (v_i,v_j) 的权 a_{ij} 表示建立城市 v_i 和城市 v_j 之间的通信线路所需的费用,则最小生成树就给出了建立通信网络最经济的方案.

最小生成树问题的数学表示如下(0-1 规划模型)：

$$\min z(y) = \sum_{(i,j)\in A} d_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{j\in V} x_{1j} \geqslant 1 \quad \text{(根至少有一条边连接到其他点)}$$

$$\sum_{j\in V} x_{ij} = 1, \quad i \neq 1 \quad \text{(除根以外,每个点只有一条边进入)}$$

$$x_{ij} = 0 \text{ 或 } 1, \quad i,j \in \mathbf{N} \quad \text{(决策变量,} x_{ij} = 1 \text{ 表示连接)}$$

对一般问题,用贪婪算法得到的只是近似最优解,而不能保证得到最优解. 但用贪婪方法计算最小生成树,却可以设计出保证得到最优解的算法. 下面我们介绍的 Kruskal 避圈法就是一个基于贪婪算法的最小生成树算法.

Kruskal 避圈法是 Kruskal 在 1956 年提出的最小生成树算法,它的思路很容易理解. Kruskal 算法每次选择 $n-1$ 条边,所使用的贪婪准则是:从剩下的边中选择一条不会产生回路的具有最小权重的边加入已选择的边的集合中. 注意到所选取的边若产生回路则不可能形成一棵生成树. 设 $G=(V,E)$是一个连通赋权图,$V=\{1,2,\cdots,n\}$. 避圈法将图 G 中的边按权数从小到大逐条考察,按不构成圈的原则加入到 T 中(若有选择时,不同的选择可能会导致最后生成树的权数不同),直到 $q(T)=n-1$ 为止,即 T 的边数$=G$ 的顶点数减 1 为止.

Kruskal 避圈法(以例 4.2 为例)的 MATLAB 程序代码如下：

```
n=8;A=[0 2 8 1 0 0 0 0; 2 0 6 0 1 0 0 0;8 6 0 7 5 1 2 0;1 0 7 0 0 0 9 0;
0 1 5 0 0 3 0 8;0 0 1 0 3 0 4 6;0 0 2 9 0 4 0 3;0 0 0 0 8 6 3 0];
k=1;  %记录 A 中不同正数的个数
for(i=1:n-1)for(j=i+1:n)  %此循环是查找 A 中所有不同的正数
if(A(i,j)>0)x(k)=A(i,j);  %数组 x 记录 A 中不同的正数
kk=1;  %临时变量
for(s=1:k-1)if(x(k)==x(s))kk=0;break;end;end  %排除相同的正数
k=k+kk;end;end;end
k=k-1  %显示 A 中所有不同正数的个数
for(i=1:k-1)for(j=i+1:k)  %将 x 中不同的正数从小到大排序
if(x(j)<x(i))xx=x(j);x(j)=x(i);x(i)=xx;end;end;end
T(n,n)=0;  %将矩阵 T 中所有的元素赋值为 0
q=0;  %记录加入到树 T 中的边数
for(s=1:k)if(q==n)break;end  %获得最小生成树 T,算法终止
for(i=1:n-1)for(j=i+1:n)if(A(i,j)==x(s))T(i,j)=x(s);T(j,i)=x(s);
%加入边到树 T 中
TT=T;  %临时记录 T
while(1)pd=1;  %砍掉 TT 中所有的树枝
for(y=1:n)kk=0;
for(z=1:n)if(TT(y,z)>0)kk=kk+1;zz=z;end;end  %寻找 TT 中的树枝
```

```
if(kk==1)TT(y,zz)=0;TT(zz,y)=0;pd=0;end;end  %砍掉 TT 中的树枝
if(pd)break;end;end  %已砍掉了 TT 中所有的树枝
pd=0;  %判断 TT 中是否有圈
for(y=1:n-1)for(z=y+1:n)if(TT(y,z)>0)pd=1;break;end;end;end
if(pd)T(i,j)=0;T(j,i)=0;  %假如 TT 中有圈
else q=q+1;end;end;end;end;end
%显示近似最小生成树 T,程序结束
```

另外 Prim 在 1957 年提出另一种最小生成树算法——Prim 算法,这种算法特别适用于边数相对较多,即比较接近于完全图的图.此算法是按逐个将顶点连通的步骤进行的,它只需采用一个顶点集合.这个集合开始时是空集,以后将已连通的顶点陆续加入到集合中去,到全部顶点都加入到集合中了,就得到所需的生成树.

设 $G=(V,E)$ 是一个连通赋权图,$V=\{1,2,\cdots,n\}$. 构造 G 的一棵最小生成树的 Prim 算法的过程是:首先从图的任一顶点起进行,将它加入集合 S 中,$S=\{1\}$,然后作如下的贪婪选择,从与之相关联的边中选出权值 a_{ij} 最小的一条作为生成树的一条边,此时满足条件 $i\in S, j\in V-S$,并将该 j 加入集合 S 中,表示这两个顶点已被所选出的边连通了.如此进行下去,直到全部顶点都加入到集合 S 中.在这个过程中选取到的所有边恰好构成 G 的一棵最小生成树.

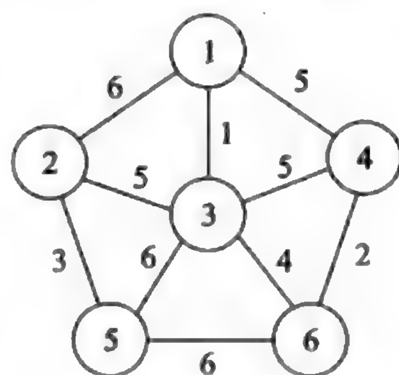


图 4.4　6 个点的连通赋权图

由于 Prim 算法中每次选取的边两端总是一个已连通顶点和一个未连通顶点,故这个边选取后一定能将该未连通点连通而又保证不会形成回路.例如,对于图 4.4 中的赋权图,按 Prim 算法选取边的过程如图 4.5 所示.

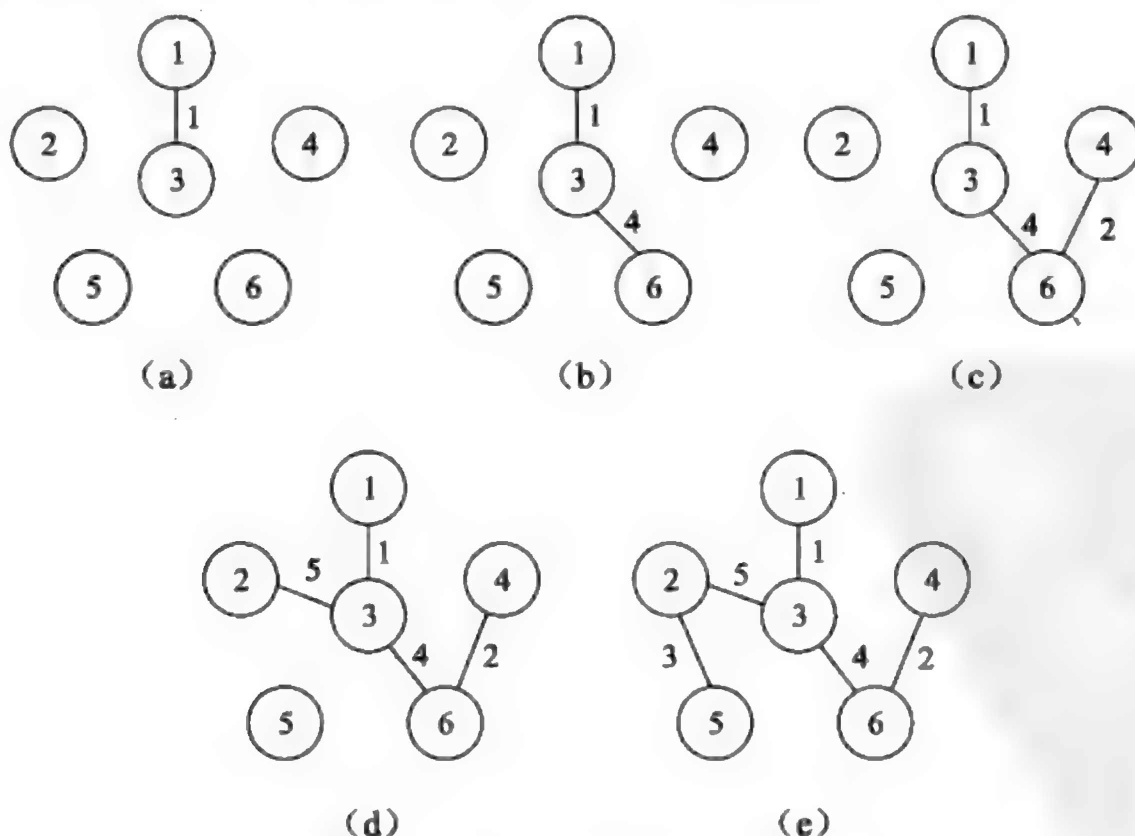


图 4.5　按 Prim 算法选取边的步骤

Prim 算法程序如下：

```
function [T,c]=tree_prim(a,n)
T=[];c=0;v=1;R=2:n;
for j=2:n
      b(1,j-1)=1;
      b(2,j-1)=j;
      b(3,j-1)=a(1,j);
end

while size(T,2)<n-1
     [q,i]= min(b(3,:));
     T(:,size(T,2)+1)=b(:,i);
     c=c+b(3,i);
     v=b(2,i);
     temp=find(R==b(2,i));
     R(temp)=[];b(:,i)=[];
     for j=1:length(R)
          d=a(v,b(2,j));
          if d<b(3,j)
               b(1,j)=v;b(3,j)=d;
          end
     end
end
```

评论　最小生成树的使用是非常广泛的，具有计算复杂度低的优点.

4.4　TSP 问题

下面讨论一下图论中著名的 TSP 问题.

我们以一个简单的例题来介绍这类问题的常用解法：

例 4.3　(旅行推销员问题(travel salesman problem ，简称 TSP 问题)，又称货郎担问题、推销员问题、旅行商问题等)设 A,B,C,D,E 5 个城市($n=5$)之间的路程由如下距离矩阵 $\boldsymbol{W}$ 给出. 假设一个推销员想从城市 A 出发，问是否存在一个行程安排，使得他能不重复地遍历所有城市后回到这个城市，而且所走路程最短. 试编程输出访问城市次序.

$$\boldsymbol{W}=\begin{bmatrix} \infty & 14 & 2 & 16 & 2 \\ 14 & \infty & 25 & 1 & 3 \\ 2 & 25 & \infty & 9 & 9 \\ 16 & 1 & 9 & \infty & 6 \\ 2 & 3 & 9 & 6 & \infty \end{bmatrix}$$

旅行推销员问题,用图论来说即“已给一个 n 个点的完全图,每条边都有一个长度 d_{ij},求总长度最短的经过每个顶点正好一次的一条封闭回路. 这个问题可分为对称旅行商问题($d_{ij}=d_{ji}$, $\forall i,j=1,2,\cdots,n$)和非对称旅行商问题($d_{ij}\neq d_{ji}$, $\forall i,j=1,2,\cdots,n$). 旅行商问题的数学模型为

$$\min z = \sum_{(i,j)\in A} d_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{j\in V} x_{1j} \geqslant 1 \quad \text{(根至少有一条边连接到其他点)}$$

$$\sum_{i\in V} x_{ij} = 1, \quad j\neq 1 \quad \text{(除根以外,每个点只有一条边进入)}$$

$$x_{ij} = 0 \text{ 或 } 1, \quad i,j\in \mathbf{N}$$

旅行商问题是一个典型的组合优化问题,TSP 问题显然是 NP 问题. 因为如果任意给出一个行程安排,可以很容易算出旅行总路程. 但是,要想知道一条总路程最小的行程是否存在,在最坏情况下,必须检查所有可能的旅行安排. 随着 n 的加大,可能的路径数目与城市数目 n 是成指数型增长的,这将是个天文数字.

如果只有 3 个城市 A,B 和 C,互相之间都有道路相连,而且起始城市是任意的,则有 6 种访问每个城市的次序:ABC,ACB,BAC,BCA,CAB,CBA. 如果有 4 个城市(图 4.6),则有 24 种次序,可以用阶乘来表示:$4!=4\times3!=4\times3\times2\times1=24$. 如果有 5 个城市,则有 $5!=5\times4!=120$,类似地,有 $6!=720$ 等. 即使用计算机来计算,这种急剧增长的可能性的数目也远远超过计算资源的处理能力.

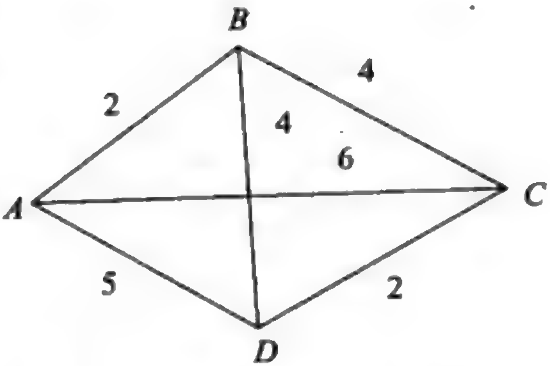


图 4.6　4 个城市的完全图

问题的关键是某些事情在实践中行不通. Cook 评论:“如果有 100 个城市,需要求出 100! 条路线的费用,没有哪一台计算机能够胜任这一任务. 打个比方,让太阳系中所有的电子以它旋转的频率来计算,就算太阳烧尽了也算不完.

1998 年,科学家们成功地解决了美国 13509 个城市之间的 TSP 问题,2001 年又解决了德国 15112 个城市之间的 TSP 问题. 但这一工程代价也是巨大的,共使用了美国 Rice 大学和 Princeton 大学之间网络互连的、由速度为 500 MHz 的 Compaq EV6 Alpha 处理器组成的 110 台计算机,所有计算机花费的时间之和为 22.6 年.

下面我们考虑用贪婪算法来处理这个问题:将城市间的距离从小到大排列有

$$d_{24}(1), d_{13}(2), d_{15}(2), d_{25}(3), d_{45}(6), d_{35}(9), d_{34}(9), d_{12}(14), d_{14}(16), d_{23}(25)$$

由于是 5 个城市,环绕一圈为 5 条边,贪婪算法求解此问题的过程是从最小边开始,依次从小到大取边加入到回路边集中,但在将 1 条边加入时不能使 1 顶点的度数超过 3,也不能形成小回路.

此问题解的过程如下:

$d_{24}(1)$;

$d_{24}(1)+d_{13}(2)$;

$d_{24}(1)+d_{13}(2)+d_{15}(2)$;

$d_{24}(1)+d_{13}(2)+d_{15}(2)+d_{25}(3)$;

$d_{24}(1)+d_{13}(2)+d_{15}(2)+d_{25}(3)+[d_{45}(6)]$;

(下标中 5 出现了 3 次,顶点 5 有三条边相连,$d_{35}(6)$放弃)

$d_{24}(1)+d_{13}(2)+d_{15}(2)+d_{25}(3)+[d_{35}(9)]$;

(下标中 5 出现了 3 次,顶点 5 有三条边相连,$d_{35}(6)$放弃)

$d_{24}(1)+d_{13}(2)+d_{15}(2)+d_{25}(3)+d_{34}(9)$.

得到一条回路 $v_1 \to v_3 \to v_4 \to v_2 \to v_5 \to v_1$是最佳的回路.

例 4.4 设有 6 个城市,其坐标分别为 $a(0,0)$,$b(4,3)$,$c(1,7)$,$d(15,7)$,$e(15,4)$,$f(18,0)$. 6 个城市间的距离矩阵为

$$D=\begin{bmatrix} \infty & 5 & 7.07 & 16.55 & 15.52 & 18 \\ 5 & \infty & 5 & 11.7 & 11.01 & 14.32 \\ 7.07 & 5 & \infty & 14 & 14.32 & 18.38 \\ 16.55 & 11.7 & 14 & \infty & 3 & 7.62 \\ 15.52 & 11.01 & 14.32 & 3 & \infty & 5 \\ 18 & 14.32 & 18.38 & 7.62 & 5 & \infty \end{bmatrix}$$

用贪婪算法,先将任两城市间的连线距离按从小到大的次序排列,然后从中逐个选择. 但有两种情况的连线应舍弃:①使任一城市的度数(连线数)超过 2 的连线必须舍弃;②在得到经过所有点的回路前就形成小回路的连线必须舍弃.

距离按从小到大的次序排列:

$D_{de}(3)$, $D_{ab}(5)$, $D_{bc}(5)$,

$D_{ef}(5)$, $D_{ae}(7.07)$, $D_{df}(7.62)$,

$D_{be}(11.01)$, $D_{bd}(11.7)$, $D_{cd}(14)$,

$D_{bf}(14.32)$, $D_{ce}(14.32)$, $D_{ae}(15.52)$,

$D_{ad}(14.55)$, $D_{af}(18.38)$, $D_{ef}(18.38)$

按贪婪算法原则,其选择过程如下:

D_{de};

$D_{de}+D_{ab}$;

$D_{de}+D_{ab}+D_{bc}$;

$D_{de}+D_{ab}+D_{bc}+D_{ef}$;

$D_{de}+D_{ab}+D_{bc}+D_{ef}+[D_{ae}]$; (形成小回路,舍弃)

$D_{de}+D_{ab}+D_{bc}+D_{ef}+[D_{df}]$; (形成小回路,舍弃)

$D_{de}+D_{ab}+D_{bc}+D_{ef}+[D_{be}]$; ($b$ 顶点度数超过 2,舍弃)

$D_{de}+D_{ab}+D_{bc}+D_{ef}+[D_{bd}]$；（$b$ 顶点度数超过 2，舍弃）

$D_{de}+D_{ab}+D_{bc}+D_{ef}+D_{cd}$；

$D_{de}+D_{ab}+D_{bc}+D_{ef}+D_{cd}+[D_{bf}]$；（$b$ 顶点度数超过 2，舍弃）

$D_{de}+D_{ab}+D_{bc}+D_{ef}+D_{cd}+[D_{ce}]$；（$c$，$e$ 顶点度数超过 2，舍弃）

$D_{de}+D_{ab}+D_{bc}+D_{ef}+D_{cd}+[D_{ae}]$；（$e$ 顶点度数超过 2，舍弃）

$D_{de}+D_{ab}+D_{bc}+D_{ef}+D_{cd}+[D_{ae}]$；（$e$ 顶点度数超过 2，舍弃）

$D_{de}+D_{ab}+D_{bc}+D_{ef}+D_{cd}+[D_{ad}]$；（$d$ 顶点度数超过 2，舍弃）

$D_{de}+D_{ab}+D_{bc}+D_{ef}+D_{cd}+D_{af}$；　得到 1 条回路

最后得到的回路如图 4.7 所示的结果，总长度为 50. 不过，这不是此问题的最优解，此问题的最优解为图 4.8 所示的路径（可以用分枝定界等方法求得），总长度为 48.39. 用贪婪算法得到的结果同最优解相比只多了 3.3%.

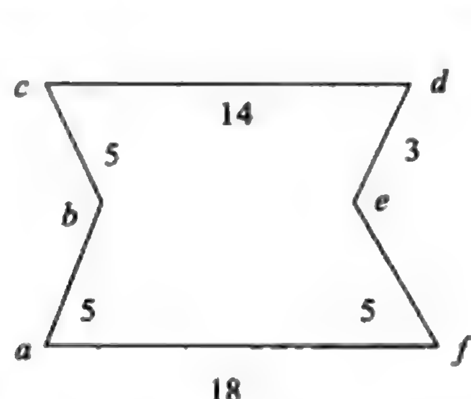


图 4.7　按贪婪算法所得的回路

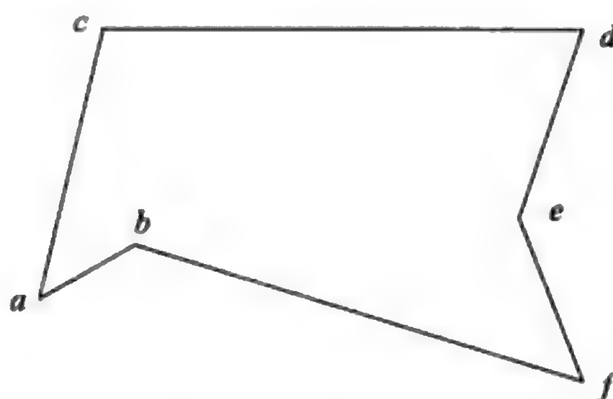


图 4.8　最优回路

TSP 问题在图论上被称为哈密顿回路问题，即指找从一点出发不重复地走过所有的结点，最后又回到原出发点的路径的问题，我们下面还跟大家介绍欧拉回路问题，它是指从一点出发不重复地走过所有的边，最后又回到原出发点的路径的问题. 两个问题的不同点在于：哈密顿回路问题是访问每个结点一次，而欧拉回路问题是访问每条边一次. 对图 G 是否存在欧拉回路，欧拉已给出充分必要条件，而对图 G 是否存在哈密顿回路至今仍未找到满足该问题的充分必要条件. 欧拉回路问题也称为中国邮递员问题. 中国数学家管梅谷于 1960 年首先研究并给出算法.

中国邮递员问题（著名图论问题之一）：邮递员从邮局出发送信，要求对辖区内每条街，都至少通过一次，再回邮局. 在此条件下，怎样选择一条最短路线？

算法：找总权数最小的欧拉回路，遍历图.

欧拉回路与欧拉图：通过图 G 的每条边一次仅且一次，而且走遍每个结点的通路（回路），就是欧拉通路（回路）. 存在欧拉回路的图就是欧拉图. 欧拉回路要求边不能重复，结点可以重复. 笔不离开纸，不重复地走完所有的边，且走过所有结点，就是所谓的一笔画. 欧拉图的所有顶点度数为偶数，且为连通图.

欧拉图或通路的判定：

(1) 无向连通图 G 是欧拉图;G 不含奇数度结点(G 的所有结点度数为偶数).

(2) 非平凡连通图 G 含有欧拉通路;G 最多有两个奇数度的结点.

(3) 连通有向图 D 含有有向欧拉回路;D 中每个结点的入度等于出度.

连通有向图 D 含有有向欧拉通路;D 中除两个结点外,其余每个结点的入度等于出度,且此两点满足:一个端点入度比出度大 1;一个端点入度比出度小 1.

例 4.5 (两只蚂蚁比赛问题)两只蚂蚁甲、乙分别处在图 4.9 中的顶点 a,b 处,并设图中各边长度相等. 甲提出同乙比赛:从它们所在顶点出发,走过图中所有边最后到达顶点 c 处. 如果它们速度相同,问谁最先到达目的地?

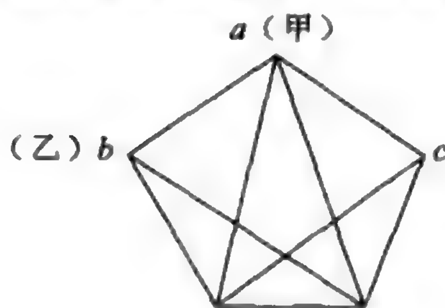


图 4.9 蚂蚁比赛

图 4.9 中有两个奇度顶点 b,c,因此存在从 b 到 c 的欧拉通路,蚂蚁乙走到 c 只要走一条欧拉通路,边数为 9,而蚂蚁甲要想走完图中所有边到达 c,至少要先走一条边到达 b,再走一条欧拉通路,故它至少要走 10 条边到达 c,所以乙必胜.

评论 最短路径问题是现实生活中十分常见的问题,其难点在于普通算法计算复杂度太大而导致难以进行计算,故在做这类问题时,要充分考虑计算复杂度的问题,要在验证算法可行的情况下进行最优解的贪婪搜索.

4.5 着色问题

已知图 $G=(V,E)$,对图 G 的所有顶点进行着色时,要求相邻的两顶点的颜色不一样,问至少需要几种颜色? 这就是所谓的顶点着色问题.

若对图 G 的所有边进行着色时,要求相邻的两条边的颜色不一样,问至少需要几种颜色? 这就是所谓的边着色问题.

这些问题的提出是有实际背景的. 值得注意的是,着色模型中的图是无向图. 对于顶点着色问题,若是有限图,也可转化为有限简单图. 而边着色问题可以转化为顶点着色问题.

例如,物资储存问题:一家公司制造 n 种化学制品 $A_1,A_2,\cdots,A_n$,其中有些化学制品若放在一起可能产生危险,如引发爆炸或产生毒气等,称这样的化学制品是不相容的. 为安全起见,在储存这些化学制品时,不相容的不能放在同一储存室内. 问至少需要多少个储存室才能存放这些化学制品?

作图 G,用顶点 $v_1,v_2,\cdots,v_n$ 分别表示 n 种化学制品,顶点 v_i 与 v_j 相邻,当且仅当化学制品 A_i 与 A_j 不相容.

于是储存问题就化为对图 G 的顶点着色问题,对图 G 的顶点最少着色数目便是最少需要的储存室数.

又如,时间表问题:现有 m 个工作人员 $x_1,x_2,\cdots,x_m$,操作 n 种设备 $y_1,y_2,\cdots,$

y_n. 设工作人员 x_i 使用设备 y_j 的时间为 a_{ij}，假定使用的时间均以单位时间计算，矩阵 $\mathbf{A}=(a_{ij})_{n\times n}$ 称为工作要求矩阵. 假定每一个工作人员在同一时间只能使用一种设备，某一种设备在同一时间里只能为一个工作人员所使用. 问应如何合理安排，使得在尽可能短的时间里满足工作人员的要求？这就是要求在工作人员与设备之间找到一个对应. 在同一时间内，工作人员 x_i 使用设备 y_j 对应一条从 x_i 到 y_j 的边，问题变为对所得的二部图 G 的边着色问题，有相同的颜色的边可以安排在同一时间里.

这些都可以转化为着色模型. 下面我们来介绍着色的方法.

对图 $G=(V,E)$ 的顶点进行着色所需最少的颜色数目用 $\chi(G)$ 表示，称为图 G 的色数.

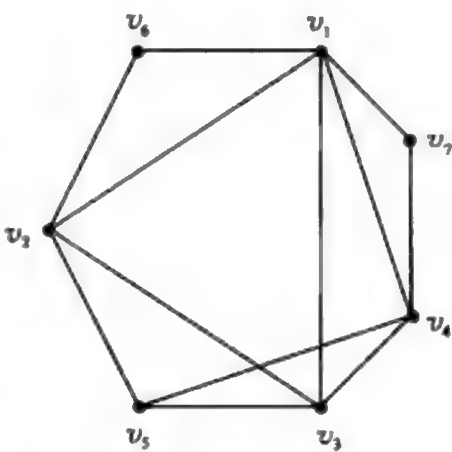


图 4.10　着色问题

定理 4.1　若图 $G=(V,E)$，$d=\max\{d(v)\mid v\in V\}$，则 $\chi(G)\leqslant d+1$.

这个定理给出了色数的上界. 着色算法目前还没有找到最优算法.

例 4.6　将图 4.10 中顶点进行着色，找到着色最少所需颜色数.

模型建立　引入 0-1 变量 x_{ik}，当 v_i 着第 k 种颜色时，$x_{ik}=1$；否则，$x_{ik}=0$. 设颜色种数为 x，建立如下模型：

$$\min x$$

$$\text{s.t.}\begin{cases}\sum\limits_{k=1}^{\Delta+1} x_{ik}=1, & i=1,2,\cdots,n\\ x_{ik}+x_{jk}\leqslant 1, & v_iv_j\in E\\ x\geqslant \sum\limits_{k=1}^{\Delta+1} kx_{ik}, & i=1,2,\cdots,n\\ x_{ik}=0\text{ 或 }1, & i=1,2,\cdots,n;k=1,2,\cdots,\Delta+1\\ x\geqslant 0\end{cases}$$

下面给出一种近似算法——最大度数优先的 Welsh-Powell 算法. 这个算法也是一个贪婪算法，算法给出了一个较好的着色方法，但不是最有效的方法，即所用的颜色数不一定是最少的，但在许多问题上，它还是有效的.

最大度数优先的 Welsh-Powell 算法如下：

设 $G=(V,E)$，$V=\{v_1,v_2,\cdots,v_n\}$，且不妨假设 $d(v_1)\geqslant d(v_2)\geqslant\cdots\geqslant d(v_n)$，$c_1$，$c_2$，$\cdots$，$c_n$ 为 n 种不同的颜色.

① 令有序集 $C_i=\{c_1,c_2,\cdots,c_n\}$ $(i=1,2,\cdots,n;j=1)$. 转向②.

② 给 v_j 着 C_j 的第一个颜色 C_{j1}. 当 $j=n$ 时，停；否则，转向③.

③ $\forall k>j$，若 v_k 和 v_j 相邻，则令 $C_k=C_k\backslash\{C_{j1}\}$. $j=j+1$，转向②.

利用 Welsh-Powell 算法，我们也可以写出上述问题的 MATLAB 程序：

```
function[a1,b1]=sortd(n,a,b)  %排序
[m,n]=size(a);
for h=1:n-1
  for j=1:n-h
   if a(j)<a(j+1)
       t1=a(j);a(j)=a(j+1);a(j+1)=t1;
      t2=b(j);b(j)=b(j+1);b(j+1)=t2;
   end
  end
end
a1=a;
b1=b;
function welsh(n,e)
totalcolor=0;  %e为邻接矩阵
temp=zeros(1,n);  %临时矩阵
for (i=1:n) temp=e(i,:);t(i)=sum(temp);temp=0;q(i)=i;c(i)=i;end
%t记录vi的度—邻接矩阵行和,q记录顶点标号,c:颜色集
[t,q]=sortd(n,t,q);  %按度排序
[m,n]=size(e);
bepaint=zeros(m,n);  %初始化着色矩阵
for i=1:n bepaint(i,:)=c; end
i=1;
     for j=1:n  %顶点-行
        pcivj=min(bepaint(j,:));bepaint(j,i)=pcivj;  %求行最小值
          for k=j+1:n
               if(e(j,k)==1)bepaint(k,i)=bepaint(k,i)+inf;end
          end
           pcivj=0;i=i+1;
     end
bepaint
```

命令窗口输入：

```
e=[0 1 1 1 0 1 1
   1 0 1 0 1 1 0
   1 1 0 1 1 0 0
   1 0 1 0 1 0 1
   0 1 1 1 0 0 0
```

```
    1 1 0 0 0 0 0
    1 0 0 1 0 0 0];
welsh(7,e)
```

评论 着色问题也是现实生活中常见的一类问题，在使用时应当注意根据计算规模的大小选择合适的算法，在求解小规模问题时适宜采用线性规划模型进行全局最优解的求解，而在求解大规模问题时应当采用 Welsh-Powell 算法进行求解，以保证算法的可行性.

4.6 最大流问题

定义 4.5 设 $G=(V,E)$为有向图，在 V 中指定一点称为发点(记为 v_s)，另一点称为收点(记为 v_t)，其余点称为中间点. 对每一条边 $v_iv_j \in E$，对应一个非负实数 C_{ij}，称为它的容量. 这样的 G 称为容量网络，简称网络，记为 $G=(V,E,C)$.

定义 4.6 网络 $G=(V,E,C)$中任一条边 v_iv_j 有流量 f_{ij}，称集合 $f=\{f_{ij}\}$为网络 G 上的一个流.

满足下述条件的流 f 称为可行流：

(1) (限制条件)对每一边 v_iv_j，有 $0 \leqslant f_{ij} \leqslant C_{ij}$.

(2) (平衡条件)对于中间点 v_k，有 $\sum f_{ik} = \sum f_{kj}$，即中间点 v_k 的输入量等于输出量.

如果 f 是可行流，则对收、发点 v_t，v_s 有 $\sum f_{si} = \sum f_{jt} = W_f$，即从 v_s 点发出的物质总量等于 v_t 点输入的量. W_f 称为网络流 f 的总流量.

上述概念可以这样来理解，如 G 是一个运输网络，则发点 v_s 表示发送站，收点 v_t 表示接收站，中间点 v_k 表示中间转运站，可行流 f_{ij} 表示某条运输线上通过的运输量，容量 C_{ij} 表示某条运输线能承担的最大运输量，W_f 表示运输总量.

可行流总是存在的，如所有边的流量 $f_{ij}=0$ 就是一个可行流(称为零流).

所谓最大流问题就是在容量网络中，寻找流量最大的可行流.

实际问题中，一个网络会出现下面两种情况：

(1) 发点和收点都不止一个. 解决的方法是再虚设一个发点 v_s 和一个收点 v_t，发点 v_s 到所有原发点边的容量都设为无穷大，所有原收点到收点 v_t 边的容量都设为无穷大.

(2) 网络中除了边有容量外，点也有容量. 解决的方法是将所有有容量的点分成两个点，如点 v 有容量 C_v，将点 v 分成两个点 v' 和 v''，令 $C(v'v'')=C_v$.

最大流问题的数学模型表示如下：

$$\max\ z = v_f$$

$$\text{s.t.}\quad \sum_{\substack{j\in V\\(i,j)\in A}} f_{ij} - \sum_{\substack{j\in V\\(j,i)\in A}} f_{ji} = \begin{cases} v_f, & i = s \\ -v_f, & i = t \\ 0, & i \neq s,t \end{cases}$$

$$0 \leqslant f_{ij} \leqslant C_{ij},\quad (i,j) \in A$$

例 4.7 现需要将城市 s 的石油通过管道运送到城市 t，中间有 4 个中转站 v_1，v_2，v_3，v_4，城市与中转站的连接以及管道的容量如图 4.11 所示，求从城市 s 到城市 t 的总的最大运输量.

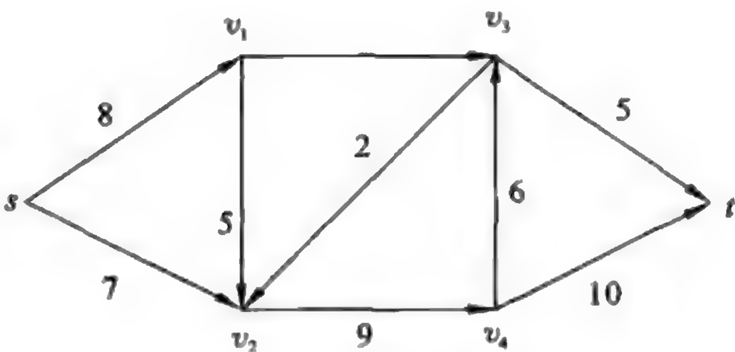


图 4.11 管道运输

最大流的 Ford-Fulkerson 标号算法如下：

(1) 标号过程.

① 给发点 v_s以标号$(+,\delta_s)$，$\delta_s=+\infty$.

② 选择一个已标号的点 x，对于 x 的所有未给标号的邻接点 y，按下列规则处理：

当 $yx\in E$ 且 $f_{yx}>0$ 时，令 $\delta_y=\min\{f_{yx},\delta_x\}$，并给 y 以标号$(x-,\delta_y)$；

当 $xy\in E$ 且 $f_{xy}<C_{xy}$时，令 $\delta_y=\min\{C_{xy}-f_{xy},\delta_x\}$，并给 y 以标号$(x+,\delta_y)$.

③ 重复②直到收点 v_t被标号或不再有点可标号时为止. 若 v_t得到标号，说明存在一条可增广链，转(2) 调整过程；若 v_t未得到标号，标号过程已无法进行时，说明 f 已经是最大流.

(2) 调整过程.

④ 决定调整量 $\delta=\delta_{vt}$，令 $u=v_t$.

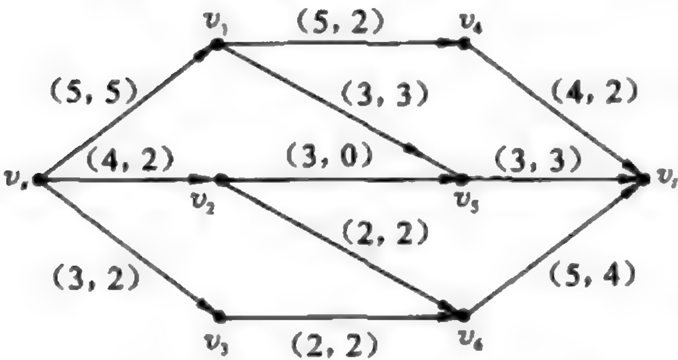


图 4.12 求所示网络最大流

⑤ 若 u 点标号为$(v+,\delta_u)$，则以 $f_{vu}+\delta$ 代替 f_{vu}；若 u 点标号为$(v-,\delta_u)$，则以 $f_{vu}-\delta$ 代替 f_{vu}.

⑥ 若 $v=v_s$，则去掉所有标号转(1)重新标号；否则令 $u=v$，转⑤.

算法终止后，令已有标号的点集为 S，则割集(S,S^c)为最小割，从而 $W_f=C(S,S^c)$.

例 4.8 求图 4.12 所示网络的最大流.

利用 Ford-Fulkerson 标号法求最大流算法的 MATLAB 程序代码如下：

```
n=8;C=[0  5  4  3  0  0  0  0
       0  0  0  0  5  3  0  0
       0  0  0  0  0  3  2  0
```

```
        0    0  0  0  0  0  2  0
        0    0  0  0  0  0  0  4
        0    0  0  0  0  0  0  3
        0    0  0  0  0  0  0  5
        0    0  0  0  0  0  0  0];  %弧容量
for(i=1:n)for(j=1:n)f(i,j)=0;end;end  %取初始可行流 f 为零流
for(i=1:n)No(i)=0;d(i)=0;end  %No,d 记录标号
while(1)
    No(1)=n+1;d(1)=Inf;  %给发点 vs 标号
    while(1)pd=1;  %标号过程
        for(i=1:n)if(No(i))  %选择一个已标号的点 vi
            for(j=1:n)if(No(j)==0&f(i,j)<C(i,j))
            %对于未给标号的点 vj,当 vivj 为非饱和弧时
                    No(j)=i;d(j)=C(i,j)-f(i,j);pd=0;
                    if(d(j)>d(i))d(j)=d(i);end
                elseif(No(j)==0&f(j,i)>0)
                %对于未给标号的点 vj,当 vjvi 为非零流弧时
                    No(j)=-i;d(j)=f(j,i);pd=0;
                    if(d(j)>d(i))d(j)=d(i);end;end;end;end;end
        if(No(n)|pd)break;end;end  %若收点 vt 得到标号或者无法标号,终止标号过程
    if(pd)break;end  %vt 未得到标号,f 已是最大流,算法终止
    dvt=d(n);t=n;  %进入调整过程,dvt 表示调整量
    while(1)
        if(No(t)>0)f(No(t),t)=f(No(t),t)+dvt;  %前向弧调整
        elseif(No(t)<0)f(No(t),t)=f(No(t),t)-dvt;end  %后向弧调整
        if(No(t)==1)for(i=1:n)No(i)=0;d(i)=0;end;break;end
        %当 t 的标号为 vs 时,终止调整过程
        t=No(t);end;end;  %继续调整前一段弧上的流 f
wf=0;for(j=1:n)wf=wf+f(1,j);end  %计算最大流量
f  %显示最大流
wf  %显示最大流量
No  %显示标号,由此可得最小割,程序结束
```

评论 在处理结点个数较少的问题时,可采用线性规划模型,在解决节点数目较多的问题时,则应当采用 Ford-Fulkerson 标号算法.

4.7 最小费用流问题

这里我们要进一步探讨不仅要使网上的流达到最大,或者达到要求的预定值,而

且还要使运输流的费用是最小的，这就是最小费用流问题.

最小费用流问题的一般提法：已知网络 $G=(V,E,C)$，每条边 $v_iv_j\in E$ 除了已给容量 C_{ij} 外，还给出了单位流量的费用 $b_{ij}(\geqslant 0)$. 所谓最小费用流问题就是求一个总流量已知的可行流 $f=\{f_{ij}\}$ 使得总费用 $b(f)=\sum\limits_{v_iv_j\in E}b_{ij}f_{ij}$ 最小.

特别地，当要求 f 为最大流时，此问题即为最小费用最大流问题.

最小费用流问题的数学表示如下：

$$\min z=\sum_{(i,j)\in A}b_{ij}f_{ij}$$

$$\text{s.t.}\quad \sum_{\substack{j\in V\\(i,j)\in A}}f_{ij}-\sum_{\substack{j\in V\\(j,i)\in A}}f_{ji}=\begin{cases}v_f, & i=s\\ -v_f, & i=t\\ 0, & i\neq s,t\end{cases}$$

$$0\leqslant f_{ij}\leqslant C_{ij},\quad (i,j)\in A$$

(1) 设网络 $G=(V,E,C)$，取初始可行流 f 为零流，求解最小费用流问题的迭代步骤如下.

① 构造有向赋权图 $G_f=(V,E_f,F)$，对于任意的 $v_iv_j\in E$，E_f，F 的定义如下：

当 $f_{ij}=0$ 时，$v_iv_j\in E_f$，$F(v_iv_j)=b_{ij}$；

当 $f_{ij}=C_{ij}$ 时，$v_jv_i\in E_f$，$F(v_jv_i)=-b_{ij}$；

当 $0<f_{ij}<C_{ij}$ 时，$v_iv_j\in E_f$，$F(v_iv_j)=b_{ij}$，$v_jv_i\in E_f$，$F(v_jv_i)=-b_{ij}$. 然后转向②.

② 求出有向赋权图 $G_f=(V,E_f,F)$ 中发点 v_s 到收点 v_t 的最短路 μ，若最短路 μ 存在，转向③；否则 f 是所求的最小费用最大流，停止.

③ 增流. 同求最大流的方法一样，重述如下：

令 $\delta_{ij}=\begin{cases}C_{ij}-f_{ij}, & v_iv_j\in\mu^+,\\ f_{ij}, & v_iv_j\in\mu^-,\end{cases}$ $\delta=\min\{\delta_{ij}\mid v_iv_j\in\mu\}$，重新定义流 $f=\{f_{ij}\}$ 为

$$f_{ij}=\begin{cases}f_{ij}+\delta, & v_iv_j\in\mu^+\\ f_{ij}-\delta, & v_iv_j\in\mu^-\\ f_{ij}, & \text{其他}\end{cases}$$

如果 W_f 大于或等于预定的流量值，则适当减少 δ 值，使 W_f 等于预定的流量值，那么 f 是所求的最小费用流，停止；否则转向①.

(3) 求解含有负权的有向赋权图 $G=(V,E,F)$ 中某一点到其他各点最短路的 Ford 算法.

当 $v_iv_j\in E$ 时，记 $w_{ij}=F(v_iv_j)$，否则取 $w_{ii}=0$，$w_{ij}=+\infty(i\neq j)$. v_1 到 v_i 的最短路长记为 $\pi(i)$，v_1 到 v_i 的最短路中 v_i 的前一个点记为 $\theta(i)$. Ford 算法的迭代步骤：

① 赋初值 $\pi(1)=0$，$\pi(i)=+\infty$，$\theta(i)=i$ $(i=2,3,\cdots,n)$.

② 更新 $\pi(i)$,$\theta(i)$. 对于 $i=2,3,\cdots,n$ 和 $j=1,2,\cdots,n$,如果 $\pi(i)<\pi(j)+w_{ji}$,则令 $\pi(i)=\pi(j)$,$\theta(i)=j$.

③ 终止判断. 若所有的 $\pi(i)$ 都无变化,停止;否则转向②.

在算法的每一步中,$\pi(i)$ 都是从 v_1 到 v_i 的最短路长度的上界. 若不存在负长回路,则从 v_1 到 v_i 的最短路长度是 $\pi(i)$ 的下界,经过 $n-1$ 次迭代后 $\pi(i)$ 将保持不变. 若在第 n 次迭代后 $\pi(i)$ 仍在变化时,说明存在负长回路.

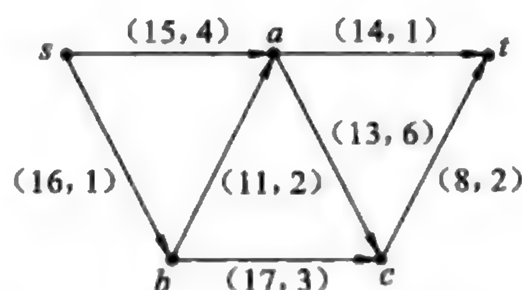


图 4.13 求 s 到 t 的最小费用最大流

例 4.9 在图 4.13 所示运输网络上,求 s 到 t 的最小费用最大流,括号内为 (C_{ij},b_{ij}).

求最小费用最大流算法的 MATLAB 程序代码如下:

```
n=5;C=[0    15   16   0   0
       0    0    0    13  14
       0    11   0    17  0
       0    0    0    0   8
       0    0    0    0   0];  %弧容量
b=[0    4   1    0    0
   0    0   0    6    1
   0    2   0    3    0
   0    0   0    0    2
   0    0   0    0    0];  %弧上单位流量的费用
wf=0;wf0=Inf;  %wf 表示最大流量,wf0 表示预定的流量值
for(i=1:n)for(j=1:n)f(i,j)=0;end;end  %取初始可行流 f 为零流
while(1)
    for(i=1:n)for(j=1:n)if(j~=i)a(i,j)=Inf;end;end;end  %构造有向赋权图
    for(i=1:n)for(j=1:n)if(C(i,j)>0&f(i,j)==0)a(i,j)=b(i,j);
        elseif(C(i,j)>0&f(i,j)==C(i,j))a(j,i)=-b(i,j);
        elseif(C(i,j)>0)a(i,j)=b(i,j);a(j,i)=-b(i,j);end;end;end
    for(i=2:n)p(i)=Inf;s(i)=i;end  %用 Ford 算法求最短路,赋初值
    for(k=1:n)pd=1;  %求有向赋权图中 vs 到 vt 的最短路
        for(i=2:n)for(j=1:n)if(p(i)>p(j)+a(j,i))p(i)=p(j)+a(j,i);
          s(i)=j;pd=0;end;end;end
        if(pd)break;end;end  %求最短路的 Ford 算法结束
    if(p(n)==Inf)break;end
    %不存在 vs 到 vt 的最短路,算法终止.注意在求最小费用最大流时构造有向赋权图
     中不会含负权回路,所以不会出现 k=n
```

```
        dvt=Inf;t=n;  %进入调整过程,dvt 表示调整量
        while(1)  %计算调整量
            if(a(s(t),t)>0)dvtt=C(s(t),t)-f(s(t),t);  %前向弧调整量
            elseif(a(s(t),t)<0)dvtt=f(t,s(t));end  %后向弧调整量
            if(dvt>dvtt)dvt=dvtt;end
            if(s(t)==1)break;end  %当 t 的标号为 vs 时,终止计算调整量
            t=s(t);end  %继续调整前一段弧上的流 f
       pd=0;if(wf+dvt>=wf0)dvt=wf0-wf;pd=1;end
        %如果最大流量大于或等于预定的流量值
       t=n;while(1)  %调整过程
            if(a(s(t),t)>0)f(s(t),t)=f(s(t),t)+dvt;  %前向弧调整
            elseif(a(s(t),t)<0)f(t,s(t))=f(t,s(t))-dvt;end  %后向弧调整
            if(s(t)==1)break;end  %当 t 的标号为 vs 时,终止调整过程
            t=s(t);end
       if(pd)break;end  %如果最大流量达到预定的流量值
       wf=0;for(j=1:n)wf=wf+f(1,j);end;end  %计算最大流量
   zwf=0;for(i=1:n)for(j=1:n)zwf=zwf+b(i,j)*f(i,j);end;end  %计算最小费用
   f  %显示最小费用最大流
   wf  %显示最小费用最大流量
   zwf  %显示最小费用,程序结束
```

运行结果如下:

```
f=
     0     6    16     0     0
     0     0     0     0    14
     0     8     0     8     0
     0     0     0     0     8
     0     0     0     0     0
wf=22
zwf=110
```

结果分析 最小费用最大流量为 22,最小费用为 110. f 矩阵为最小费用最大流.

4.8 二部图的匹配及应用

定义 4.7 设 X,Y 都是非空有限顶点集,且 $X\cap Y=\varnothing$,$E\subset\{xy|x\in X,y\in Y\}$,称 $G=(X,Y,E)$ 为二部图. 如果 X 中的每个点都与 Y 中的每个点邻接,则称 $G=(X,Y,E)$ 为完备二部图. 若 $F:E\to\mathbf{R}^+$,则称 $G=(X,Y,E,F)$ 为二部赋权图.

定义 4.8 若图 $G=(V,E)$，$M\subset E$，若 M 中任意两条边在 G 中均不邻接，则称 M 是 G 的一个匹配.

定义 4.9 若匹配 M 的某条边与点 v 关联，则称 M 饱和点 v，并且称 v 是 M 的饱和点，否则称 v 是 M 的非饱和点.

4.8.1 最大匹配

定义 4.10 设 M 是图 G 的一个匹配，如果 G 的每一个点都是 M 的饱和点，则称 M 是完美匹配；如果 G 中没有另外的匹配 M_0，使 $|M_0|>|M|$，则称 M 是最大匹配. 每个完美匹配都是最大匹配，反之不一定成立.

定义 4.11 设 M 是图 G 的一个匹配，其边在 $E\backslash M$ 和 M 中交错出现的路，称为 G 的一条 M-交错路. 起点和终点都不是 M 的饱和点的 M-交错路，称为 M-增广路.

定理 4.2 G 的一个匹配 M 是最大匹配的充要条件是 G 不包含 M-增广路.

定理 4.3 设 $G=(X,Y,E)$ 为二部图，则

(1) G 存在饱和 X 的每个点的匹配的充要条件是对任意 S，有 $|N(S)|\geqslant|S|$，其中，$N(S)=\{v|u\in S,v$ 与 u 相邻$\}$.

(2) G 存在完美匹配的充要条件是对任意 $S\subset X$ 或 $S\subset Y$ 有 $|N(S)|\geqslant|S|$.

工作安排问题之一

给 n 个工作人员 $x_1,x_2,\cdots,x_n$ 安排 n 项工作 $y_1,y_2,\cdots,y_n$. n 个工作人员中每个人能胜任一项或几项工作，但并不是所有工作人员都能从事任何一项工作. 例如，x_1 能做 y_1,y_2 工作，x_2 能做 y_2,y_3,y_4 工作等.

这样便提出一个问题，对所有的工作人员能不能都分配一件他所能胜任的工作？

构造一个二部图 $G=(X,Y,E)$，其中 $X=\{x_1,x_2,\cdots,x_n\}$，$Y=\{y_1,y_2,\cdots,y_n\}$，并且当且仅当工作人员 x_i 胜任工作 y_j 时，x_i 与 y_j 才相邻.

于是，问题转化为求二部图的一个完美匹配. 因为 $|X|=|Y|$，所以完美匹配即为最大匹配.

求二部图 G 的最大匹配的算法(匈牙利算法)，其基本思想是：从 G 的任意匹配 M 开始，对 X 中所有 M 的非饱和点，寻找 M-增广路. 若不存在 M-增广路，则 M 为最大匹配；若存在 M-增广路 P，则将 P 中 M 与非 M 的边互换得到比 M 多一边的匹配 M_1，再对 M_1 重复上述过程.

设 $G=(X,Y,E)$ 为二部图，其中 $X=\{x_1,x_2,\cdots,x_n\}$，$Y=\{y_1,y_2,\cdots,y_n\}$. 任取 G 的一初始匹配 M(如任取 $e\in E$，则 $M=\{e\}$ 是一个匹配).

① 令 $S=\varnothing$，$T=\varnothing$，转向②.

② 若 M 饱和 $X\backslash S$ 的所有点，则 M 是二部图 G 的最大匹配；否则，任取 M 的非饱和点 $u\in X\backslash S$，令 $S=S\cup\{u\}$，转向③.

③ 记 $N(S)=\{v|u\in S,uv\in E\}$. 若 $N(S)=T$，转向②；否则取 $y\in N(S)\backslash T$. 若 y

是 M 的饱和点，转向④；否则转向⑤.

④ 设 $xy \in M$，则令 $S=S\cup\{x\}$，$T=T\cup\{y\}$，转向③.

⑤ $u-y$ 路是 M-增广路，设为 P，并令 $M=M\oplus P$，转向①. 其中，$M\oplus P=M\cup P\backslash M\cap P$，是对称差.

由于计算 M-增广路 P 比较麻烦，因此将迭代步骤改为：

① 将 X 中 M 的所有非饱和点(不是 M 中某条边的端点)都给以标号 0 和标记 *，转向②.

② 若 X 中所有有标号的点都已去掉了标记 *，则 M 是 G 的最大匹配；否则任取 X 中一个既有标号又有标记 * 的点 x_i，去掉 x_i 的标记 *，转向③.

③ 找出在 G 中所有与 x_i 邻接的点 y_j(即 $x_iy_j\in E$)，若所有这样的 y_j 都已有标号，则转向②，否则转向④.

④ 对与 x_i 邻接且尚未给标号的 y_j 都给定标号 i. 若所有的 y_j 都是 M 的饱和点，则转向⑤；否则逆向返回. 即由其中 M 的任一个非饱和点 y_j 的标号 i 找到 x_i，再由 x_i 的标号 k 找到 y_k，最后由 y_t 的标号 s 找到标号为 0 的 x_s 时结束，获得 M-增广路 $x_sy_t,\cdots,x_iy_j$，记 $P=\{x_sy_t,\cdots,x_iy_j\}$，重新记 M 为 $M\oplus P$，转向①.

⑤ 将 y_j 在 M 中与之邻接的点 x_k(即 $x_ky_j\in M$)，给以标号 j 和标记 *，转向②.

例 4.10 求图 4.14 中所示的二部图 G 的最大匹配.

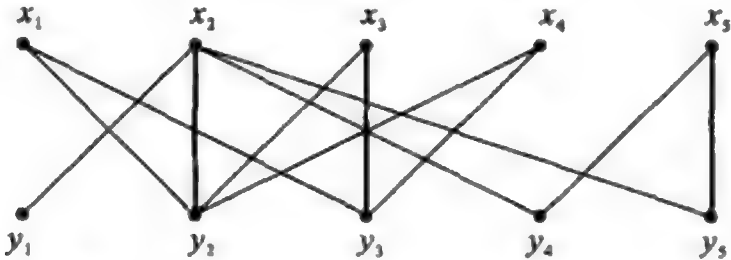


图 4.14 求所示二部图的最大匹配

匈牙利算法的 MATLAB 程序代码如下：

```
m=5;n=5;A=[0   1   1   0   0
           1   1   0   1   1
           0   1   1   0   0
           0   1   1   0   0
           0   0   0   1   1];
M(m,n)=0;
for(i=1:m)for(j=1:n)if(A(i,j))M(i,j)=1;break;end;end  %求初始匹配M
    if(M(i,j))break;end;end  %获得仅含一条边的初始匹配M
while(1)
    for(i=1:m)x(i)=0;end  %将记录X中点的标号和标记*
    for(i=1:n)y(i)=0;end  %将记录Y中点的标号和标记*
    for(i=1:m)pd=1;  %寻找X中M的所有非饱和点
```

```
            for(j=1:n)if(M(i,j))pd=0;end;end
            if(pd)x(i)= -n-1;end;end
%将 X 中 M 的所有非饱和点都给以标号 0 和标记*,程序中用 n+1 表示 0 标号,标号为负
 数时表示标记*
        pd=0;
        while(1)xi=0;
            for(i=1:m)if(x(i)<0)xi=i;break;end;end
%假如 X 中存在一个既有标号又有标记*的点,则任取 X 中一个既有标号又有标记*的点 xi
            if(xi==0)pd=1;break;end
%假如 X 中所有有标号的点都已去掉了标记*,算法终止
            x(xi)=x(xi)*(-1);  %去掉 xi 的标记*
            k=1;
            for(j=1:n)if(A(xi,j)&y(j)==0)y(j)=xi;yy(k)=j;k=k+1;end;end
%对与 xi 邻接且尚未给标号的 yj 都给以标号 i
            if(k>1)k=k-1;
                for(j=1:k)pdd=1;
                    for(i=1:m)if(M(i,yy(j)))x(i)=-yy(j);pdd=0;break;end;end
%将 yj 在 M 中与之邻接的点 xk(即 xkyj∈M),给以标号 j 和标记*
                    if(pdd)break;end;end
                if(pdd)k=1;j=yy(j);  %yj 不是 M 的饱和点
                    while(1)P(k,2)=j;P(k,1)=y(j);j=abs(x(y(j)));
%任取 M 的一个非饱和点 yj,逆向返回
                        if(j==n+1)break;end
%找到 X 中标号为 0 的点时结束,获得 M-增广路 P
                        k=k+1;end
                    for(i=1:k)if(M(P(i,1),P(i,2)))M(P(i,1),P(i,2))=0;
%将匹配 M 在增广路 P 中出现的边去掉
                        else M(P(i,1),P(i,2))=1;end;end
%将增广路 P 中没有在匹配 M 中出现的边加入到匹配 M 中
                    break;end;end;end
        if(pd)break;end;end  %假如 X 中所有有标号的点都已去掉了标记*,算法终止
M  %显示最大匹配 M,程序结束
```

运行结果如下:

```
M=
    0    1    0    0    0
    1    0    0    0    0
    0    0    1    0    0
    0    0    0    0    0
    0    0    0    1    0
```

结果分析 M矩阵为最大匹配的匹配方式. 匹配方式为(x_1,y_2),(x_2,y_1),(x_3,y_3),(x_5,y_4).

4.8.2 最佳匹配

定义 4.12 设$G=(X,Y,E,F)$为完备的二部赋权图,若$L:X\cup Y\to\mathbf{R}^+$满足:对任意$x\in X,y\in Y,L(x)+L(y)\geqslant F(xy)$,则称$L$为$G$的一个可行点标记,记相应的生成子图为$G_L=(X,Y,E_L,F)$,其中$E_L=\{xy\in E|L(x)+L(y)=F(xy)\}$.

定理 4.4 设L是完备的二部赋权图$G=(X,Y,E,F)$的可行点标记,若M^*是G_L的完美匹配,则M^*是G的最佳匹配.(权数最大的匹配)

工作安排问题之二

给n个工作人员$x_1,x_2,\cdots,x_n$安排n项工作$y_1,y_2,\cdots,y_n$. 如果每个工作人员工作效率不同,要求工作分配的同时考虑总效率最高.

我们构造一个二部赋权图$G=(X,Y,E,F)$,其中$X=\{x_1,x_2,\cdots,x_n\}$,$Y=\{y_1,y_2,\cdots,y_n\}$,$F(x_i y_j)$为工作人员x_i胜任工作y_j时的工作效率,则问题转化为求二部赋权图G的最佳匹配. 在求G的最佳匹配时,总可以假设G为完备二部赋权图. 若x_i与y_j不相邻,可令$F(x_i y_j)=0$. 同样地,还可虚设点x或y,使$|X|=|Y|$. 如此就将G转化为完备二部赋权图,而且不会影响结果.

1. 可行点标号求最佳匹配

设$G=(X,Y,E,F)$为完备的二部赋权图,L是其一个初始可行点标记,通常取

$$\begin{cases}L(x)=\max\{F(xy)|y\in Y\}, & x\in X\\ L(y)=0, & y\in Y\end{cases}$$

M是G_L的一个匹配.

① 若X的每个点都是M的饱和点,则M是最佳匹配;否则取M的非饱和点$u\in X$,令$S=\{u\}$,$T=\varnothing$,转向②.

② 记$N_L(S)=\{v|u\in S,uv\in E_L\}$. 若$N_L(S)=T$,则$G_L$没有完美匹配,转向③;否则转向④.

③ 调整可行点标记,计算:

$$a_L=\min\{L(x)+L(y)-F(xy)|x\in S,y\in Y\backslash T\}$$

由此得新的可行顶点标记

$$H(v)=\begin{cases}L(v)-a_L, & v\in S\\ L(v)+a_L, & v\in T\\ L(v), & \end{cases}$$

否则,令$L=H$,$G_L=G_H$,重新给出G_L的一个匹配M,转向①.

④ 取$y\in N_L(S)\backslash T$,若y是M的饱和点,转向⑤;否则,转向⑥.

⑤ 设$xy\in M$,则令$S=S\cup\{x\}$,$T=T\cup\{y\}$,转向②.

⑥ 在 G_L 中的 $u-y$ 路是 M-增广路，记为 P，并令 $M=M\oplus P$，转向①.

由于程序源码过长，读者请登陆华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目下载学习利用可行点标记求最佳匹配算法的 MATLAB 程序.

2. 矩阵覆盖法求最佳匹配

(1) 求等价分配矩阵.

(2) 求独立零元(非同列同行的零)，画上框.

(3) 最优判别：达到 n 个独立零元，停止.

(4) 求覆盖线：

① 封锁没有画框零元的行，封锁就打√；

② 在封锁行中未画框零元的列也封锁；

③ 在封锁列中画框零元的行也封锁；

④ 未封锁行与封锁列画上覆盖线.

(5) 调节分配矩阵：在未覆盖元中选取最大元 k，未覆盖行加 $|k|$，覆盖列减 $|k|$，转(2).

由于程序源码过长，读者请登陆华中数学建模网 http://www.shumo.cn/大学数学实验栏目下载学习矩阵覆盖法的 MATLAB 程序.

例 4.11 (关于灾情巡视问题的分析与解答，节选)图 4.15 为某县的乡(镇)、村公路网示意图，公路边的数字为该路段的公里数. 夏天该县遭受水灾. 为考察灾情，组织自救，县领导决定，带领有关部门负责人到全县各乡(镇)、村巡视. 巡视路线为从县政府所在地出发，走遍各乡(镇)、村，又回到县政府所在地.

(1) 若分三组(路)巡视，试设计总路程最短且各组尽可能均衡的巡视路线.

(2) 在上述关于 T，t 和 V 的假定下，如果巡视人员足够多，完成巡视的最短时间是多少？给出在这种最短时间完成巡视的要求下，你认为最佳的巡视路线.

分析 本题是一类图上点的遍历性问题，即要用若干条回路覆盖图上所有的顶点，并使某些指标达到最优.

一种通常的思路即将题目中的求最佳巡视路线的问题转化为图论中的旅行商(哈密顿回路)的问题. 将题目所给的地图视为一个赋权无向连通图. 关于灾情巡视时的分组实际上就是将原图进行顶点集合的划分而得到一些子图，然后从总体上得到一些回路使得这些回路可以经过顶点集合中的每一个元素. 可是在解题的过程中我们可以发现，如果要求所有的推销员回路的路程最短时会出现某一条回路和另一条回路相差较大，因而会使一组巡视完后等待另一组较长的时间，因此这种较短路程并没有起到节约时间的作用. 而从节约时间的角度考虑，我们应该使分组中的每一条回路的路程相差不大，于是有我们定义的一个均衡度来描述. 综合考虑总路程最短和各组巡视路线尽可能均衡的要求，我们可以近似找到一个较好的分组方案.

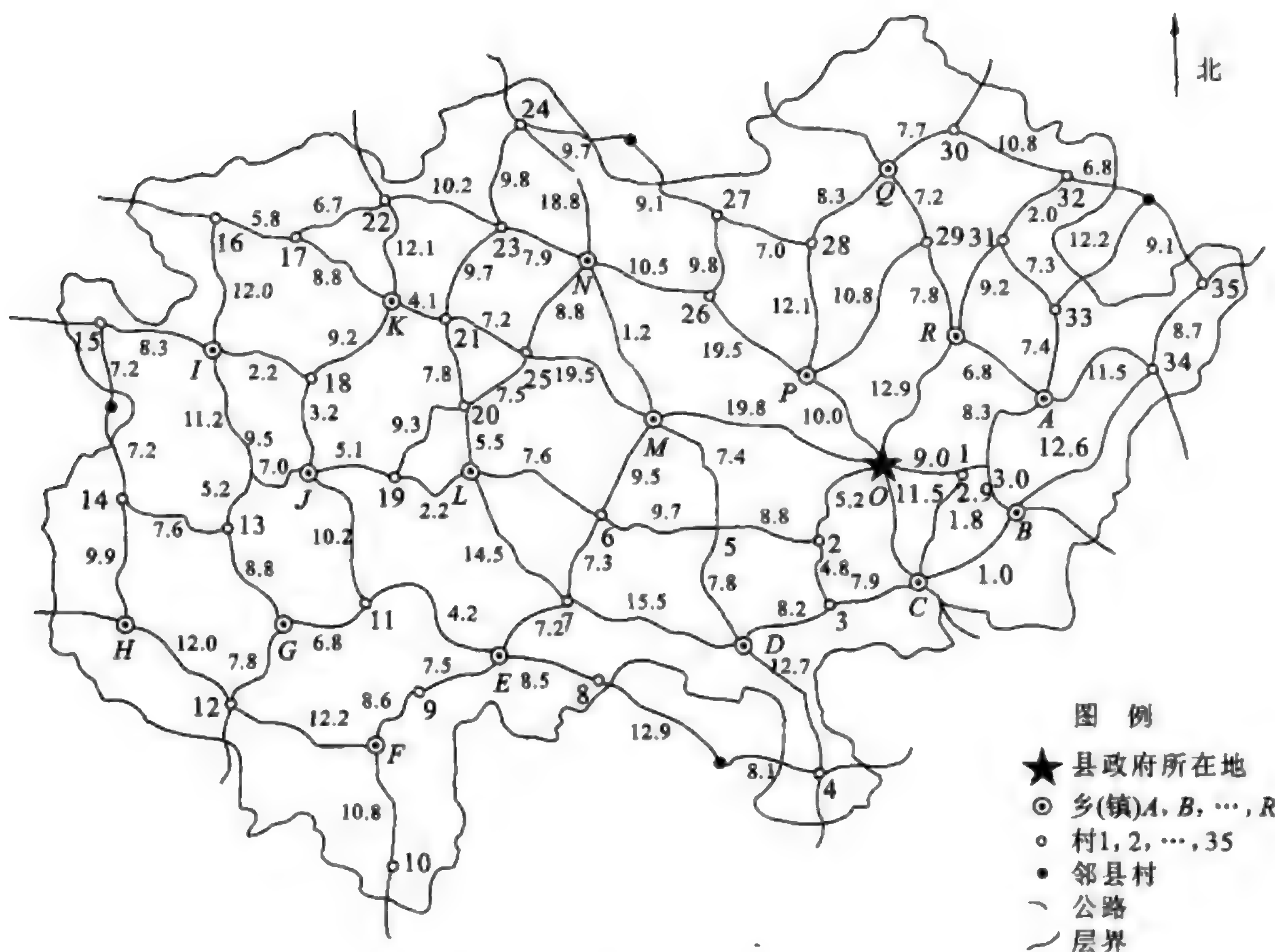


图 4.15　灾情巡视问题

分组巡视的最短路是多旅行商问题，旅行商问题有两种形式：过每点一次仅且一次；过每点至少一次. 理论上，两点间距离满足三角不等式时，两种形式等价；否则应作修改将后者化为前者，才能用旅行商问题的典型算法，实际上本题只需注意个别处即可.

简单介绍模型的建立与求解：本题要求在某个给定的县的乡、村公路网中，寻找从县政府所在地(图中的 O 点)出发，走遍各乡、村，又回到县政府所在地，使得总路程以及巡视时间尽量少. 我们可以对公路网图进行处理，每个乡或村视为图中的一个结点，各乡或村之间的公路视为图中对应结点间的边，各条公路的长度视为对应边上的权，于是公路网就转化为图论中的加权网络图，问题就变为一个图论的问题了，即在给定的加权网络图中寻找从 O 出发，行遍所有顶点至少一次再回到 O 点，使得总权最小.

(1) 若分三组巡视，设计总路程最短且各组尽可能均衡的巡视路线. 这个问题是一个多个推销员的最佳推销员回路的问题. 即要对加权图 G 中的顶点集 V 进行划分，将其分为 3 份，于是得到 $G_i=G(V_i,E_i)$ $(i=1,2,3)$. 用避圈法给出了 G 的最小生成树如图 4.16 所示.

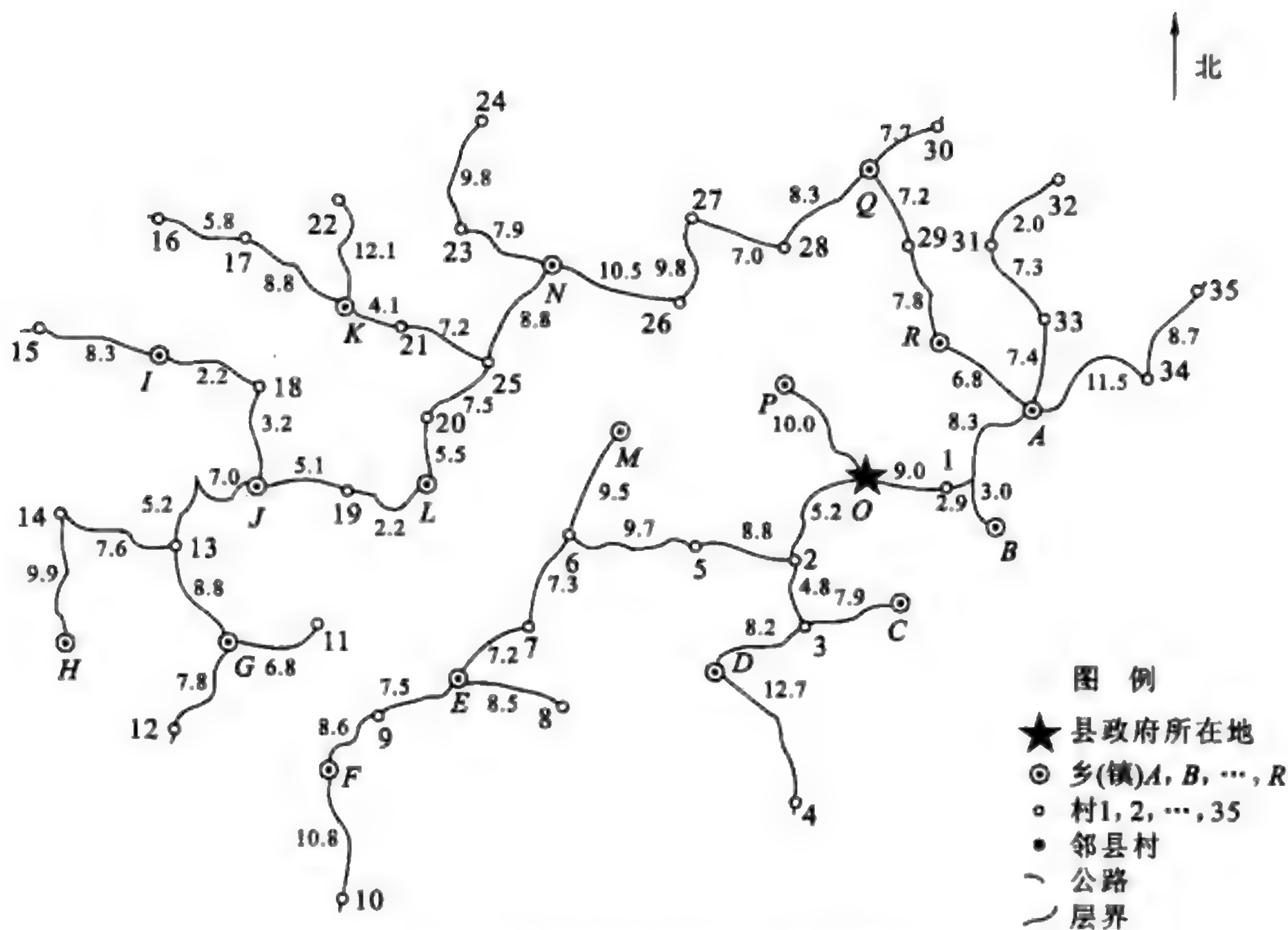


图 4.16　灾情巡视路线的最小生成树

其中最小生成树的边权总和为 424.8 km. 我们需要对上面的顶点进行划分，可以发现图 4.16 可以按照各个顶点的聚集方式及方位分成三组. 我们按照单一行遍某一地区的原则进行原图上的寻路工作，先划出顶点，就是运用中国邮递员问题的算法添加路线边，得到一个邮递员回路，然后寻找局部最优解，然后考虑总体，进行局部调整和优化，现给出下面的划分方案：

第一组为 O-1-B-A-34-35-33-31-32-30-Q-28-27-24-23-N-26-P-29-R-O，总长度为 205.1 km；

第二组为 O-M-25-21-K-22-17-16-I-15-I-18-J-19-20-L-6-5-2-O，总长度为 160.7 km；

第三组为 O-2-3-D-7-E-11-G-13-14-H-12-F-10-F-9-E-8-4-D-3-C-O，总长度为 209.9 km；

巡视总路程为 575.7 km.

$$\alpha=\frac{209.9-160.7}{209.9}=23\%$$

α 比较大，我们需要进行一些调整，发现改变个别点后可以使均衡程度得到较好的水平，但是需要增加巡视总路程.

第一组为 O-1-B-A-34-35-33-31-32-30-Q-28-27-24-23-N-26-P-29-R-O，总

长度为 205.4 km；

第二组为 O-M-25-21-K-22-17-16-I-15-14-13-J-18-J-19-20-L-6-5-2-O，总长度为 187.3 km；

第三组为 O-2-3-D-7-E-11-G-12-H-12-F-10-F-9-E-8-4-D-3-C-O，总长度为 203.4 km.

$$\alpha=\frac{205.4-187.3}{205.4}=8.9\%$$

$\alpha\leqslant 10\%$，均衡程度比较好. 可以采用这种分组方法.

(2) 如果巡视组数已定，要求尽快完成巡视，试讨论 T,t,V 的变化对于最佳巡视路线的影响.

如果分组为 3 时我们有如下的巡视时间的表达式：

$$T_i=N_i\times T+n_i\times t+l_i\div V,\quad i=1,2,3$$

由上面的式子我们可以发现：

① 当 N_i,n_i,l_i 的数目完全相同时，T,t,V 的改变对于最佳巡视路线没有影响.

② 我们将 T 和 t 做一些处理，将它们统一为时间参数，令 $T=\theta t$；于是可以将上面的公式写为

$$T_i=(N_i\times\theta+n_i)\times t+l_i\div V,\quad i=1,2,3$$

如果 V 不变，t 变化，这时有：若 t 变化幅度比较小，那么对于最佳巡视路线的影响不大；若 t 变化幅度很大，则这时各个小组巡视时在各乡村的停留的数目应该尽可能的均衡，才能保证它的改变对于巡视路线的影响不大，否则，对于巡视路线的影响就十分明显了，所以必须对巡视路线进行调整.

如果 t 不变，V 变化，这时同样有：若 V 变化幅度比较小，那么对于最佳巡视路线的影响不大；若 V 变化幅度很大，则这时各个小组巡视时 l_i 应该保证比较均衡，这样才能保证它的改变对于巡视路线的影响不大，否则，对于巡视路线的影响就十分明显了，所以必须对巡视路线进行调整.

模型的推广 类似这类关于图论的哈密顿圈和最佳推销员回路的问题至今还没有一个能够通用的解决算法，但是现在已经出现了较多的近似最优解的求解算法，它们都有一些局限性，但是我们可以应用这些算法解决实际生活中的很多问题，如旅游景点的最佳方案、车辆运输时的路线选择、一般的巡视等问题，具有十分广泛的实际应用和经济价值.

模型的分析

(1) 本例应用了逐次改进算法，它是一个近似算法，计算的复杂程度属于多项式，但是求出的解不一定是最优的.

(2) 对于 T,t 和 V 的变化对最优巡视路线的影响只是给出粗略的分析，没有给出最终的表达式.

(3) 关于第一问和第二问我们只是经过有限次的分析和比较得到我们自己认为较好的分组方案.

(4) 按照我们取分组的方法,可能由于各人对于边界处理方法的不同,可以得到有较大不同的分组情况,但是经过处理后只要它们可以达到一定的评定指标时,都可以接受.

第 5 章　动态规划与排队论

5.1　动态规划

动态规划是解决多阶段决策过程最优化的一种方法，该方法是英国数学家贝尔曼(R. E. Bellman)等人在 20 世纪 50 年代提出的. 其特点在于，它可以将一个 n 维决策问题变换为几个一维最优化问题，从而一个一个地去解决. 利用动态规划方法，已成功地解决了生产管理、工程技术等方面的许多问题.

5.1.1　动态规划的最优原理及其算法

1. 求解多阶段决策过程的方法

例 5.1　最短路问题. 求图 5.1 从 A 到 B 的最短路径.

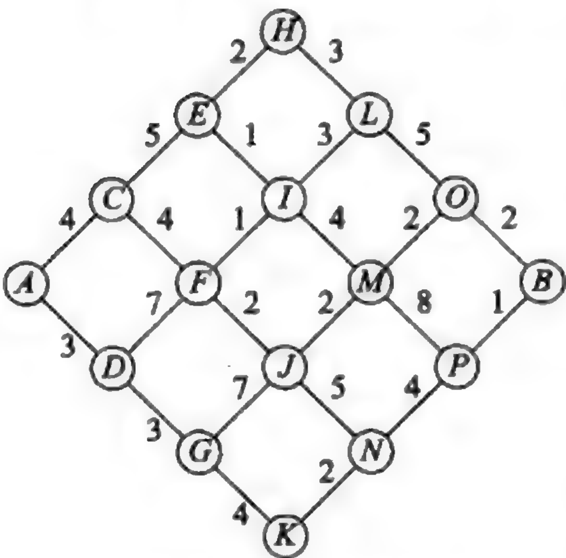


图 5.1　路径图

从图 5.1 中可知，该问题可以枚举出 20 条路径，其中最短的路径长度为 16，容易发现，最短路表现为明显的阶段性，一条从 A 到 B 的最短路径中的任何一段都是最短的.

设 S_i 表示由 i 点到 B 点的最短路径的长度，则

$$S_A=\min\{d_{AC}+S_C,d_{AD}+S_D\}$$

因此要找到 A 到 B 的最短路 S_A，必须先找到 C 到 B 的最短路 S_C 以及 D 到 B 的最短路 S_D，以此类推，从而有以下最优性原理.

最优性原理、最优策略的一部分也是最优的.

2. 动态规划的基本概念及递推公式

1) 状态

例如，最短路问题中，各个结点是状态；生产库存问题中，库存量是状态；物资分配问题中，剩余的物资量是状态.

2) 控制变量(决策变量)

在最短路问题中，走哪条路是决策变量；生产库存问题中，各阶段的产品生产量是决策变量；物资分配问题中，分配给每个地区的物资量是决策变量.

3) 阶段的编号与递推的方向

一般采用反向递推，所以阶段的编号也是逆向的. 当然也可以正向递推.

4) 动态规划的步骤

(1) 确定问题的阶段和编号.

(2) 确定状态变量. 用 s_k 表示第 k 阶段的状态变量及其值.

(3) 确定决策变量. 用 x_k 表示第 k 阶段的决策变量,并以 x_k^* 表示该阶段的最优决策.

(4) 状态转移方程. 用 $s_{k-1}=g(s_k,x_k)$ 反向编号,用 $s_{k+1}=g(s_k,x_k)$ 正向编号.

(5) 直接效果. 是指直接一步转移的效果 $d_k(s_k,x_k)$.

(6) 总效果函数. 是指某阶段某状态下到终端状态的总效果,它是一个递推公式,即

$$f_k(s_k,x_k)=h_k(d_k(s_k,x_k),f_{k-1}(s_{k-1},x_{k-1}^*))$$

其中,h_k 是一般表达形式,求当前阶段当前状态下的阶段最优总效果.

① 如最短路问题,是累加形式,此时有

$$\begin{aligned}f_k(s_k,x_k^*)&=\min_{x_k}\{d_k(s_k,x_k)+f_{k-1}(s_{k-1},x_{k-1}^*)\}\\&=d_k(s_k,x_k^*)+f_{k-1}[g(s_k,x_k^*),x_{k-1}^*]\end{aligned}$$

终端的边际效果一般为 $f_0(s_0,x_0)=0$.

② 如串联系统可靠性问题,是连乘形式,此时有

$$\begin{aligned}f_k(s_k,x_k^*)&=\max_{x_k}\{d_k(s_k,x_k)f_{k-1}(s_{k-1},x_{k-1}^*)\}\\&=d_k(s_k,x_k^*)f_{k-1}[g(s_k,x_k^*),x_{k-1}^*]\end{aligned}$$

终端的边际效果一般为 $f_0(s_0,x_0)=1$.

从第一阶段开始,利用边际效果和边界条件,可以递推到最后阶段.

5.1.2 动态规划模型举例

例 5.2 (产品生产计划安排问题)某工厂生产某种产品的月生产能力为 10 件,已知今后 4 个月的产品成本及销售量见表 5.1. 如果本月产量超过销售量时,可以存储起来备以后各月销售,一件产品的月存储费为 2 元,试安排月生产计划并做到:

(1) 保证满足每月的销售量,并规定计划期初和期末库存为零;

(2) 在生产能力允许范围内,安排每月生产量计划使产品总成本(即生产费用加存储费)最低.

表 5.1 4 个月的产品成本及销售量

月份	阶段 k	产品成本 c_k/件	月销售量 y_k/件	月初库存 s_k/件	月末库存 s_{k-1}/件
1	4	70	6	$s_4=0$	s_3
2	3	72	7	s_3	s_2
3	2	80	12	s_2	s_1
4	1	76	6	s_1	$s_0=0$

设 x_k 为第 k 阶段生产量,则直接成本为

$$d_k(s_k, x_k) = c_k x_k + 2s_k$$

状态转移公式为

$$s_{k-1} = s_k + x_k - y_k$$

总成本递推公式为：

$$f_k(s_k, x_k^*) = \min_{x_k}\{d_k(s_k, x_k) + f_{k-1}(s_{k-1}, x_{k-1}^*)\}$$

第一阶段(即第 4 个月)　由边界条件和状态转移方程：

$$s_0 = s_1 + x_1 - y_1 = s_1 + x_1 - 6 = 0$$

得

$$s_1 + x_1 = 6 \quad 或 \quad x_1 = 6 - s_1$$

估计第一阶段，即第 4 个月初库存的可能状态：$s_1 \in [0,5]$，第一阶段最优决策表见表 5.2.

表 5.2　第一阶段最优决策表

s_1	x_1^*	$f_1(s_1, x_1^*)$
0	6	456
1	5	382
2	4	308
3	3	234
4	2	160
5	1	86

第二阶段　最大可能库存量为 7 件. 由状态转移方程：

$$s_1 = s_2 + x_2 - 12 \geqslant 0 \quad 及 \quad x_2 \leqslant 10$$

知

$$s_2 \in [2,7], \qquad \min x_2 = 5$$

由阶段效果递推公式，例如，

$$f_2(2,10) = d_2(2,10) + f_1^*(0,6) = 2\times2 + 80\times10 + 456 = 1260$$

得第二阶段最优决策表见表 5.3.

表 5.3　第二阶段最优决策

s_2 \ x_2	5	6	7	8	9	10	x_2^*	$f_2(s_2, x_2^*)$
2						1260*	10	1260
3					1182*	1188	9	1182
4				1104*	1110	1116	8	1104
5			1026*	1032	1038	1044	7	1026
6		948*	954	960	966	972	6	948
7	870*	876	882	888	894	900	5	870
	$s_1=0$	$s_1=1$	$s_1=2$	$s_1=3$	$s_1=4$	$s_1=5$		

第三阶段　最大可能库存量 4 件. 由状态转移方程:

$$s_2=s_3+x_3-7\geqslant 2 \quad 及 \quad x_3\leqslant 10$$

知
$$s_3\in[0,4], \qquad \min x_3=5$$

由阶段效果递推公式,例如,

$$f_3(1,10)=d_3(1,10)+f_2^*(4,8)=2\times 1+72\times 10+1104=1826$$

得第三阶段最优决策表见表 5.4.

表 5.4　第三阶段最优决策表

s_3 \ x_3	5	6	7	8	9	10	x_3^*	$f_3(s_3,x_3^*)$
0					1908	1902*	10	1902
1				1838	1832	1826*	10	1826
2			1768	1762	1756	1750*	10	1750
3		1698	1692	1686	1680	1674*	10	1674
4	1628	1622	1616	1610	1604	1598*	10	1598
	$s_2=2$	$s_2=3$	$s_2=4$	$s_2=5$	$s_2=6$	$s_2=7$		

第四阶段　初始库存量 $s_4=0$. 由状态转移方程:

$$s_3=s_4+x_4-6\geqslant 0$$

知 $x_4\geqslant 6$,由阶段效果递推公式,例如,

$$f_4(0,6)=d_4(0,6)+f_3^*(0,10)=70\times 6+1902=2322$$

得第四阶段最优决策表见表 5.5,回溯得表 5.6.

表 5.5　第四阶段最优决策表

s_4 \ x_4	6	7	8	9	10	x_4^*	$f_4(s_4,x_4^*)$
0	2322	2316	2310	2304	2298*	10	2298
	$s_3=0$	$s_3=1$	$s_3=2$	$s_3=3$	$s_3=4$		

表 5.6　最终的最优结果

月份 k	s_k/件	x_k^*/件	y_k/件	生产费用/元	库存费/元	月总费用/元	累计费用/元
1	0	10	6	700	0	700	700
2	4	10	7	720	8	728	1428
3	7	5	12	400	14	414	1842
4	0	6	6	456	0	456	2298

例 5.3 （生产-库存管理问题（连续变量））设某厂计划全年生产某种产品 A. 其四个季度的订货量分别为 600 kg,700 kg,500 kg 和 1200 kg. 已知生产产品 A 的生产费用与产品产量的平方成正比，系数为 0.005. 厂内有仓库可存放产品，存储费为每公斤每季度 1 元. 求最佳的生产安排使年总成本最小.

四个季度为四个阶段，采用阶段编号与季度顺序一致. 设 s_k 为第 k 季初的库存量，则边界条件为 $s_1=s_5=0$. 设 x_k 为第 k 季的生产量，y_k 为第 k 季的订货量，状态转移方程为

$$s_{k+1}=s_k+x_k-y_k$$

仍采用反向递推，但注意阶段编号是正向的，目标函数为

$$f_1(x)=\min_{x_1,x_2,x_3,x_4}\sum_{i=1}^{4}(0.005x_i^2+s_i)$$

第一步（第四季度） 总效果 $f_4(s_4,x_4)=0.005x_4^2+s_4$，由边界条件有

$$s_5=s_4+x_4-y_4=0$$

解得

$$x_4^*=1200-s_4$$

将 x_4^* 代入 $f_4(s_4,x_4)$ 得

$$f_4^*(s_4)=0.005(1200-s_4)^2+s_4=7200-11s_4+0.005s_4^2$$

第二步（第三、四季度） 总效果 $f_3(s_3,x_3)=0.005x_3^2+s_3+f_4^*(s_4)$，将 $s_4=s_3+x_3-500$ 代入 $f_3(s_3,x_3)$ 得

$$\begin{aligned}f_3(s_3,x_3)&=0.005x_3^2+s_3+7200-11(x_3+s_3-500)+0.005(x_3+s_3-500)^2\\&=0.01x_3^2+0.01x_3s_3-16x_3+0.005s_3^2-15s_3+13950\end{aligned}$$

$$\frac{\partial f_3(s_3,x_3)}{\partial x_3}=0.02x_3+0.01s_3-16=0$$

解得 $x_3^*=800-0.5s_3$，代入 $f_3(s_3,x_3)$ 得

$$f_3^*(s_3)=7550-7s_3+0.0025s_3^2$$

第三步（第二、三、四季度） 总效果 $f_2(s_2,x_2)=0.005x_2^2+s_2+f_3^*(s_3)$，将 $s_3=s_2+x_2-700$ 代入 $f_2(s_2,x_2)$ 得

$$f_2(s_2,x_2)=0.005x_2^2+s_2+7550-7(x_2+s_2-700)+0.0025(x_2+s_2-700)^2$$

$$\frac{\partial f_2(s_2,x_2)}{\partial x_2}=0.015x_2+0.005(s_2-700)-7=0$$

解得 $x_2^*=700-\frac{s_2}{3}$，代入 $f_2(s_2,x_2)$ 得

$$f_2^*(s_2)=10000-6s_2+\frac{0.005s_2^2}{3}$$

注 最优阶段总效果仅是当前状态的函数，与其后的决策无关.

第四步（第一、二、三、四季度） 总效果 $f_1(s_1,x_1)=0.005x_1^2+s_1+f_2^*(s_2)$，将 $s_2=s_1+x_1-600=x_1-600$ 代入 $f_1(s_1,x_1)$ 得

$$f_1(s_1,x_1)=0.005x_1^2+s_1+10\,000-6(x_1-600)+\frac{0.005(x_1-600)^2}{3}$$

$$\frac{\partial f_1(s_1,x_1)}{\partial x_1}=\frac{0.04x_1}{3}-8=0$$

解得 $x_1^*=600$，代入 $f_1(s_1,x_1)$ 得

$$f_1^*(s_2)=11800$$

由此回溯，得最优生产-库存方案：$x_1^*=600, s_2^*=0; x_2^*=700, s_3^*=0; x_3^*=800, s_4^*=300; x_4^*=900$.

例 5.4 （资源分配问题）某公司有 9 个推销员在全国 3 个不同市场推销货物，这 3 个市场里推销人员数与收益的关系见表 5.7，试作出使总收益最大的分配方案.

表 5.7 推销人员数与收益关系表

市场 \ 推销员	0	1	2	3	4	5	6	7	8	9
1	20	32	47	57	66	71	82	90	100	110
2	40	50	60	71	82	93	104	115	125	135
3	50	61	72	84	97	109	120	131	140	150

设分配人员的顺序为市场 1,2,3，采用反向阶段编号. 设 s_k 为第 k 阶段尚未分配的人员数，边界条件为 $s_3=9$，x_k 为第 k 阶段分配的推销人员数. 仍采用反向递推，状态转移方程为

$$s_{k-1}=s_k-x_k$$

目标函数为

$$f_3(x)=\min_{x_1,x_2,x_3}\sum_{i=1}^{3}d(x_i)$$

第一阶段 给第三市场分配. s_1 有 0～9 种可能，第一阶段最优决策表见表 5.8.

表 5.8 第一阶段最优决策表

s_1 \ x_1	0	1	2	3	4	5	6	7	8	9	x_1^*	f_1^*
0	50										0	50
1	50	61									1	61
2	50	61	72								2	72
3	50	61	72	84							3	84
4	50	61	72	84	97						4	97
5	50	61	72	84	97	109					5	109

续表

x_1 \ s_1	0	1	2	3	4	5	6	7	8	9	x_1^*	f_1^*
6	50	61	72	84	97	109	120				6	120
7	0	61	72	84	97	109	120	131			7	131
8	50	61	72	84	97	109	120	131	140		8	140
9	50	61	72	84	97	109	120	131	140	150	9	150

第二阶段　给第二市场分配，见表 5.9.

表 5.9　第一阶段最优分配表

s_1	0	1	2	3	4	5	6	7	8	9
x_1^*	0	1	2	3	4	5	6	7	8	9
f_1^*	50	61	72	84	97	109	120	131	140	150

s_2 有 0～9 种可能，第二阶段最优决策表见表 5.10.

表 5.10　第二阶段最优决策表

x_2 \ s_2	0	1	2	3	4	5	6	7	8	9	x_2^*	f_2^*
0	90										0	90
1	101	100									0	101
2	112	111	110								0	112
3	124	122	121	121							0	124
4	137	134	132	132	132						0	137
5	149	147	144	143	143	143					0	149
6	160	159	157	155	154	154	154				0	160
7	171	170	169	168	166	165	165	165			0	171
8	180	181	180	180	179	177	176	176	175		1	181
9	190	190	191	191	191	190	188	187	186	185	2,3,4	191

第三阶段　给第一市场分配，见表 5.11. 由边界条件 $s_3=9$，第三阶段最优决策表见表 5.12.

表 5.11　第二阶段最优分配表

s_2	0	1	2	3	4	5	6	7	8	9
x_2^*	0	0	0	0	0	0	0	0	1	2,3,4
f_2^*	90	101	112	124	137	149	160	171	181	191

表 5.12 第三阶段最优决策表

s_3 \ x_3	0	1	2	3	4	5	6	7	8	9	x_3^*	f_3^*
9	211	213	218	217	215	208	206	202	201	200	2	218

得决策过程：$x_3^*=2, x_2^*=0, x_1^*=7, f_3^*=218$. 即市场 1 分配 2 人，市场 2 不分配，市场 3 分配 7 人.

例 5.5 （项目选择问题）某工厂预计明年有 A,B,C,D 4 个新建项目，每个项目的投资额 w_k 及其投资后的收益 v_k 见表 5.13. 投资总额为 30 万元，问如何选择项目才能使总收益最大？

表 5.13 每个项目及投资额和收益

项目	w_k	v_k
A	15	12
B	10	8
C	12	9
D	8	5

这是一类 0-1 规划问题，该问题是经典的旅行背包问题，是 NPC 问题.

上述问题的静态规划模型如下：

$$\max f(x) = \sum_k v_k x_k$$

$$\text{s. t.} \begin{cases} \sum_k w_k x_k \leqslant 30 \\ x_k = \begin{cases} 0, & \text{未投资} \\ 1, & \text{投资} \end{cases} \end{cases}$$

设项目选择的顺序为 A,B,C,D.

(1) 阶段 $k=1,2,3,4$ 分别对应 D,C,B,A 项目的选择过程.

(2) 第 k 阶段的状态 s_k 代表第 k 阶段初尚未分配的投资额.

(3) 第 k 阶段的决策变量 x_k 代表第 k 阶段分配的投资额.

(4) 状态转移方程为 $s_{k-1}=s_k-w_k x_k$.

(5) 直接效益 $d_k(s_k,x_k)=v_k$ 或 0.

(6) 总效益递推公式：

$$f_k(s_k,x_k^*)=\max_{x_k}\{d_k(s_k,x_k)+f_{k-1}(s_{k-1},x_{k-1}^*)\}$$

该问题的难点在于各阶段的状态的确定，当阶段增加时，状态数成指数增长. 下面利用决策树来确定各阶段的可能状态.

第一阶段(项目 D)　当 $s_1<8$ 时，x_1 只能取 0；$w_1=8$，$v_1=5$.

表 5.14　第一阶级可能分配的状态

s_1	x_1	$d_1(s_1,x_1)$	$s_0=s_1-w_1x_1$	$f_0(s_0,x_0^*)$	$f_1(s_1,x_1^*)$	条件
3	0	0	3	0	0	$x_4=x_2=1$
	—	—	—			$x_3=0$
5	0	0	5	0	0	$x_4=x_3=1$
	—	—	—			$x_2=0$
8	0	0	8	0		$x_3=x_2=1$
	1	5	0	0	5	$x_4=0$
15	0	0	15	0		$x_3=x_2=0$
	1	5	7	0	5	$x_4=1$
18	0	0	18	0		$x_4=x_3=0$
	1	5	10	0	5	$x_2=1$
20	0	0	20	0		$x_4=x_2=0$
	1	5	12	0	5	$x_3=1$
30	0	0	30	0		$x_4=x_3=0$
	1	5	22	0	5	$x_2=0$

第二阶段(项目 C)

表 5.15　第二阶段可能分配的状态　　$w_2=12$，$v_2=9$

s_2	x_2	$d_2(s_2,x_2)$	$s_1=s_2-w_2x_2$	$f_1(s_1,x_1^*)$	$f_2(s_2,x_2^*)$	条件
5	0	0	5	0	0	$x_4=x_3=1$
	—	—	—			
15	0	0	15	5	5	$x_3=0$
	1	9	3	0	9^*	$x_4=1$
20	0	0	20	5	5	$x_4=0$
	1	9	8	5	14^*	$x_3=1$
30	0	0	30	5	5	$x_4=x_3=0$
	1	9	18	5	14^*	

第三阶段(项目 B)

表 5.16 第三阶段可能分配的状态 $w_3=10, v_3=8$

s_3	x_3	$d_3(s_3, x_3)$	$s_2=s_3-w_3x_3$	$f_2(s_2, x_2^*)$	$f_3(s_3, x_3^*)$	条件
15	0	0	15	9	9*	$x_4=1$
	1	8	5	0	8	
15	0	0	30	14	14	$x_4=0$
	1	8	20	14	22*	

第四阶段(项目 A)

表 5.17 第四阶段可能分配的状态 $w_4=15, v_4=12$

s_4	x_4	$d_4(s_4, x_4)$	$s_3=s_4-w_4x_4$	$f_3(s_3, x_3^*)$	$f_4(s_4, x_4^*)$	条件
30	0	0	30	22	22*	
	1	12	15	9	21	

回溯得投资计划,见表 5.18.

表 5.18 投资计划表

项目	决策	投资额	直接收益 v_k
A	$x_4=0$	0	0
B	$x_3=1$	10	8
C	$x_2=1$	12	9
D	$x_1=1$	8	5
总额		30	22

例 5.6 有 A, B, C 三部机器串联生产某种产品,由于工艺技术问题,产品常出现次品. 统计结果表明,机器 A, B, C 产生次品的概率分别为 $p_A=30\%$, $p_B=40\%$, $p_C=20\%$,而产品必须经过三部机器顺序加工才能完成. 为了降低产品的次品率,决定拨款 5 万元进行技术改造,以便最大限度地提高产品的成品率指标. 现提出如下四种改进方案:

方案 1 不拨款,机器保持原状;

方案 2 加装监视设备,每部机器需 1 万元;

方案 3 加装设备,每部机器需 2 万元;

方案 4 同时加装监视及控制设备,每部机器需 3 万元.

采用各方案后,各部机器的次品率见表 5.19.

表 5.19 各机器的次品率

	A	B	C
不拨款	30%	40%	20%
拨款 1 万元	20%	30%	10%
拨款 2 万元	10%	20%	10%
拨款 3 万元	5%	10%	6%

为三台机器分配改造拨款，设拨款顺序为 A,B,C，阶段序号反向编号为 k，即第一阶段计算给机器 C 拨款的效果.

设 s_k 为第 k 阶段剩余款，则边界条件为 $s_3=5$，设 x_k 为第 k 阶段的拨款额，状态转移方程为 $s_{k-1}=s_k-x_k$，目标函数为

$$\max R=(1-P_A)(1-P_B)(1-P_C)$$

仍采用反向递推.

第一阶段 对机器 C 拨款的效果.

$$R_1(s_1,x_1)=d_1(s_1,x_1)\times R_0(s_0,x_0)=d_1(s_1,x_1)$$

表 5.20 第一阶段拨款

s_1 \ x_1	0	1	2	3	x_1^*	$R_1(s_1,x_1^*)$
0	0.8				0	0.8
1	0.8	0.9			1	0.9
2	0.8	0.9	0.9		1,2	0.9
3	0.8	0.9	0.9	0.94	3	0.94
4	0.8	0.9	0.9	0.94	3	0.94
5	0.8	0.9	0.9	0.94	3	0.94

第二阶段 对机器 B,C 拨款的效果. 由于机器 A 最多只需 3 万元，故 $s_2\geqslant 2$.

递推公式： $R_2(s_2,x_2)=d_2(s_2,x_2)\times R_1(s_1,x_1^*)$

代表数值，得 $R_2(3,2)=0.72$. 得第二阶段最优决策表，见表 5.21.

表 5.21 第二阶段最优决策表

s_1 \ x_1	x_1^*	$R_1(s_1,x_1^*)$
0	0	0.8
1	1	0.9
2	1,2	0.9

续表

s_1 \ x_1	x_1^*	$R_1(s_1,x_1^*)$
3	3	0.94
4	3	0.94
5	3	0.94

第三阶段　对机器 A,B,C 拨款的效果.

边界条件：$$s_3=5$$

递推公式：$$R_3(s_3,x_3)=d_3(s_3,x_3)\times R_2(s_2,x_2^*)$$

$$R_3(5,3)=d_3(5,3)\times R_2(2,2)=(1-0.05)\times 0.64=0.608$$

得第三阶段最优决策表，见表 5.22.

表 5.22　第三阶段最优决策表

s_3 \ x_3	0	1	2	3	x_3^*	$R_3(s_3,x_3^*)$
5	0.567	0.648	0.648	0.608	1,2	0.648

回溯：有多组最优解.

I：$x_3=1,x_2=3,x_1=1$，　　$R_3=0.8\times 0.9\times 0.9=0.648$

II：$x_3=2,x_2=2,x_1=1$，　　$R_3=0.9\times 0.8\times 0.9=0.648$

III：$x_3=2,x_2=3,x_1=0$，　　$R_3=0.9\times 0.9\times 0.8=0.648$

动态规划总结　动态规划主要有两大类：生产-库存问题和资源分配问题.

(1) 生产-库存问题.

状态转移公式：$$s_{k-1}=s_k+x_k-y_k$$

状态和控制变量：离散型或连续型.

(2) 资源分配问题.

状态转移公式：$$s_{k-1}=s_k+x_k$$

目标函数：累加或累乘.

状态和控制量：离散型或连续性.

5.2 排　队　论

排队常常是件很令人恼火的事情，尤其在我们这样的人口大国. 生活中许多时候和地方都涉及排队，如银行、ATM、医院、理发、火车售票、游乐场等. 这部分就来讨论一下生活中排队的数学模型.

5.2.1 基本概念

排队论(queuing theory),又称随机服务系统理论(random service system theory),是一门研究拥挤现象(排队、等待)的科学. 具体地说,它是在研究各种排队系统概率规律性的基础上,解决相应排队系统的最优设计和最优控制问题.

不同的顾客与服务组成了各式各样的服务系统. 顾客为了得到某种服务而到达服务系统,若不能立即获得服务则必须排队等待,待获得服务后离开系统,下面给出了各种排队系统简图,如图 5.2~5.7 所示.

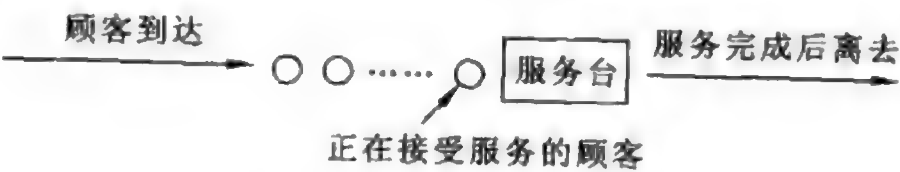


图 5.2 单服务台排队系统

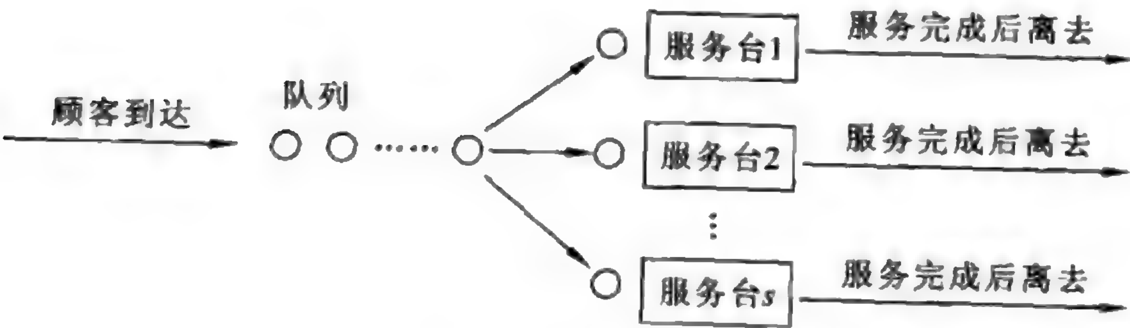


图 5.3 单队列-s 个服务台并联的排队系统

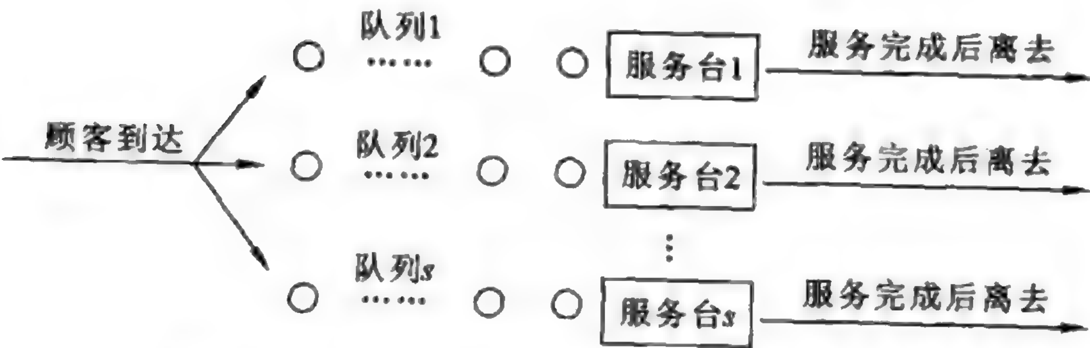


图 5.4 s 个队列-s 个服务台的并联排队系统

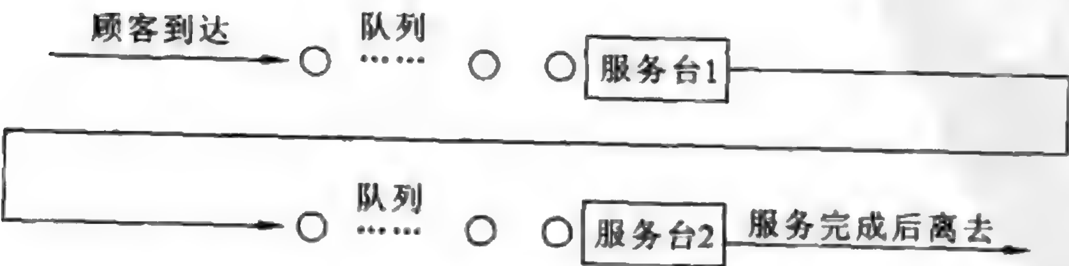


图 5.5 单队列-多个服务台的串联排队系统

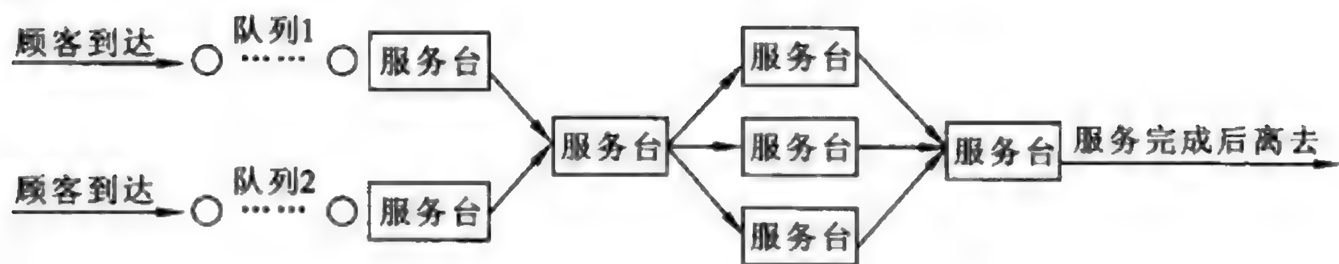


图 5.6　多队列-多服务台混联、网络系统

一般的排队系统，都可由图 5.7 中模型加以描述.

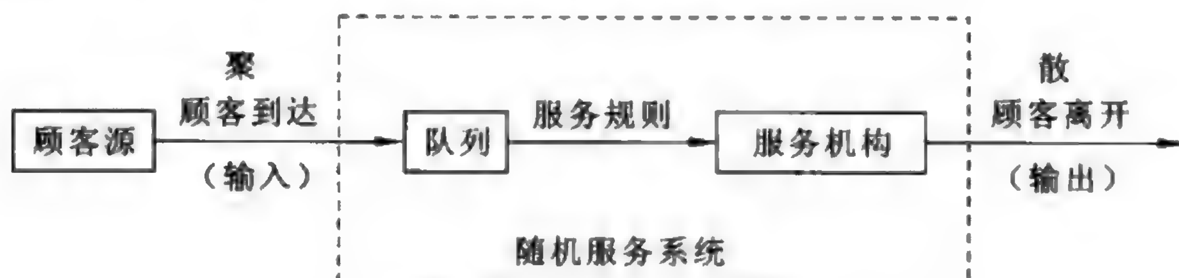


图 5.7　随机服务系统

5.2.2　排队系统的描述

1. 系统特征和基本排队过程

实际的排队系统虽然千差万别，但是它们有以下的共同特征：

(1) 有请求服务的人或物，如顾客.

(2) 有为顾客服务的人或物，即服务员或服务台.

(3) 顾客到达系统的时刻是随机的，为每一位顾客提供服务的时间是随机的，因而整个排队系统的状态也是随机的. 排队系统的这种随机性可能造成某个阶段顾客排队较长，而另外一些时候服务员（台）又空闲无事.

2. 排队系统的基本组成部分

通常，排队系统都由输入过程、服务规则和服务台三个部分组成.

(1) 输入过程. 这是指要求服务的顾客是按怎样的规律到达排队系统的过程，有时也把它称为顾客流. 一般可以从三个方面来描述一个输入过程：

① 顾客总体数，又称顾客源、输入源. 这是指顾客的来源，顾客源可以是有限的，也可以是无限的. 例如，到售票处购票的顾客总数可以认为是无限的，而某个工厂因故障待修的机床则是有限的.

② 顾客到达方式. 这是描述顾客是怎样来到系统的，他们是单个到达，还是成批到达. 病人到医院看病是顾客单个到达的，库存问题中如将生产器材进货或产品入库视为顾客，那么这种顾客则是成批到达的.

③ 顾客流的概率分布，或称相继顾客到达的时间间隔的分布. 这是在求解排队系统有关运行指标问题时，首先需要确定的指标. 这也可以理解为在一定的时间间

隔内到达 K 个顾客($K=1,2,\cdots$)的概率是多大. 顾客流的概率分布一般有定长分布、二项分布、泊松流(最简单流)、爱尔朗分布等若干种.

(2) 服务规则. 这是指服务台从队列中选取顾客进行服务的顺序. 一般可以分为损失制、等待制和混合制三大类.

① 损失制. 这是指如果顾客到达排队系统时,所有服务台都已被先来的顾客占用,那么他们就自动离开系统永不再来. 典型例子是,电话拨号后出现忙音,顾客不愿等待而自动挂断电话,如要再打,就需重新拨号,这种服务规则即为损失制.

② 等待制. 这是指当顾客来到系统时,所有服务台都不空,顾客加入排队行列等待服务,如排队等待售票、故障设备等待维修等. 等待制中,服务台在选择顾客进行服务时,常有如下四种规则:

a. 先到先服务. 按顾客到达的先后顺序对顾客进行服务,这是最普遍的情形.

b. 后到先服务. 仓库中叠放的钢材,后叠放上去的都先被领走,就属于这种情况.

c. 随机服务. 即当服务台空闲时,不按照排队序列而随意指定某个顾客服务,如电话交换台接通呼叫电话.

d. 优先权服务. 例如,老人、儿童先进车站,危重病员先就诊,遇到重要数据需要处理,计算机则立即中断其他数据的处理等,均属于此种服务规则.

③ 混合制. 这是等待制与损失制相结合的一种服务规则,一般是指允许排队,但又不允许队列无限长下去. 具体说来,大致有三种:

a. 队长有限. 当排队等待服务的顾客人数超过规定数量时,后来的顾客就自动离去,另求服务,即系统的等待空间是有限的. 例如,最多只能容纳 K 个顾客在系统中,当新顾客到达时,若系统中的顾客数(又称为队长)小于 K,则可进入系统排队或接受服务;否则,便离开系统,并不再回来. 如水库的库容是有限的,旅馆的床位是有限的.

b. 等待时间有限. 即顾客在系统中的等待时间不超过某一给定的长度 T,当等待时间超过 T 时,顾客将自动离去,并不再回来. 例如,易损坏的电子元器件的库存问题,超过一定存储时间的元器件被自动认为失效. 又如,顾客到饭馆就餐,等了一定时间后不愿再等而自动离去另找饭店用餐.

c. 逗留时间(等待时间与服务时间之和)有限. 例如,用高射炮射击敌机,当敌机飞越高射炮射击有效区域的时间为 t 时,若在这个时间内未被击落,也就不可能再被击落了.

不难注意到,损失制和等待制可看成是混合制的特殊情形,如记 s 为系统中服务台的个数,则当 $K=s$ 时,混合制即成为损失制;当 $K=\infty$时,混合制即成为等待制.

(3) 服务台情况. 服务台可以从以下三方面来描述:

① 服务台数量及构成形式. 从数量上说,服务台有单服务台和多服务台之分. 从

构成形式上看,服务台有单队-单服务台式、单队-多服务台并联式、多队-多服务台并联式、单队-多服务台串联式、单队-多服务台并串联混合式,以及多队-多服务台并串联混合式等,如图 5.2～5.7 所示.

② 服务方式. 这是指在某一时刻接受服务的顾客数,它有单个服务和成批服务两种. 例如,公共汽车一次就可装载一批乘客就属于成批服务.

③ 服务时间的分布. 一般来说,在多数情况下,对每一个顾客的服务时间是一随机变量,其概率分布有定长分布、负指数分布、k 阶爱尔朗分布、一般分布(所有顾客的服务时间都是独立同分布的)等.

5.2.3 排队系统的描述符号与分类

为了区别各种排队系统,根据输入过程、排队规则和服务机制的变化对排队模型进行描述或分类,可给出很多排队模型. 为了方便对众多模型的描述,20 世纪 50 年代肯道尔(D. G. Kendall)提出了一种目前在排队论中广泛采用的“Kendall 记号”,完整的表达方式通常用到 6 个符号并取如下固定格式:

$$A/B/C/D/E/F$$

各符号的意义为

A:顾客相继到达的间隔时间分布,常用下列符号:

M:到达过程为泊松过程或负指数分布;

D:定长输入;

E_k:k 阶爱尔朗分布;

G:一般相互独立的随机分布.

B:服务时间分布,所用符号与表示顾客到达间隔时间分布相同:

M:服务过程为泊松过程或负指数分布;

D:定长分布;

E_k:k 阶爱尔朗分布;

G:一般相互独立的随机分布.

C:服务台(员)个数:“1”表示单个服务台,“s”($s>1$)表示多个服务台.

D:系统中顾客容量限额,或称等待空间容量. 例如,系统有 K 个等待位子,则 $0<K<\infty$,当 $K=0$ 时,说明系统不允许等待,即为损失制;当 $K=\infty$时为等待制系统,此时∞般省略不写;当 K 为有限整数时,为混合制系统.

E:顾客源限额,分有限与无限两种,∞表示顾客源无限,此时一般∞也可省略不写.

F:服务规则,常用下列符号:

FCFS(first come first serve):先到先服务的排队规则;

LCFS(last come first serve):后到先服务的排队规则;

PR:优先权服务的排队规则.

例如，某排队问题为 $M/M/s/\infty/\infty/\text{FCFS}/$，则表示顾客到达间隔时间为负指数分布（泊松流）；服务时间为负指数分布；有 $s(s>1)$ 个服务台；系统等待空间容量无限（等待制）；顾客源无限，采用先到先服务规则.

某些情况下，排队问题仅用上述表达形式中的前 3 个、4 个或 5 个符号. 如不特别说明则均理解为系统等待空间容量无限，顾客源无限，先到先服务，单个服务的等待制系统.

5.2.4 排队系统的主要数量指标

研究排队系统的目的是通过了解系统运行的状况，对系统进行调整和控制，使系统处于最优运行状态. 因此，首先需要弄清系统的运行状况. 描述一个排队系统运行状况的主要数量指标有：

(1) 队长和排队长(队列长).

队长是指系统中的平均顾客数(排队等待的顾客数与正在接受服务的顾客数之和)，记为 L_s，排队长是指系统中正在排队等待服务的平均顾客数，记为 L_q.

队长和排队长一般都是随机变量，我们希望能确定它们的分布，或至少能确定它们的平均值(即平均队长和平均排队长)及有关的矩(如方差等). 队长的分布是顾客和服务员都关心的，特别是对系统设计人员来说，如果能知道队长的分布，就能确定队长超过某个数的概率，从而确定合理的等待时间.

(2) 等待时间和逗留时间.

从顾客到达时刻起到他开始接受服务为止这段时间称为等待时间，是随机变量，也是顾客最关心的指标，因为顾客通常希望等待时间越短越好. 从顾客到达时刻起到他接受服务完成止这段时间称为逗留时间，也是随机变量，同样也是顾客非常关心的. 对这两个指标的研究当然是希望能确定它们的分布，或至少能知道顾客的平均等待时间 W_q 和平均逗留时间 W_s.

(3) 忙期和闲期.

忙期是指从顾客到达空闲着的服务台起，到服务台再次成为空闲为止的这段时间，即服务台连续忙的时间. 这是个随机变量，是服务员最为关心的指标，因为它关系到服务员的服务强度. 与忙期相对的是闲期，即服务机构连续保持空闲的时间. 在排队系统中，忙期和闲期总是交替出现的.

除了上述几个基本数量指标外，还会用到其他一些重要的指标，如在损失制或系统容量有限的情况下，由于顾客被拒绝，而使服务系统受到损失的顾客损失率及服务强度等，也都是十分重要的数量指标.

(4) 一些数量指标的常用记号.

① 主要数量指标：

L 或 L_s：平均队长，即稳态系统任一时刻的所有顾客数的期望值；

L_q：平均等待队长或队列长，即稳态系统任一时刻的等待服务的顾客数的期望值；

W 或 W_s：平均逗留时间，即(在任意时刻)进入稳态系统的顾客逗留时间的期望值；

W_q：平均等待时间，即(在任意时刻)进入稳态系统的顾客等待时间的期望值.

② 其他常用数量指标：

s：系统中并联服务台的数目；

λ：平均到达率；

$1/\lambda$：平均到达间隔；

μ：平均服务率；

$1/\mu$：平均服务时间；

ρ：服务强度，即每个服务台单位时间内的平均服务时间，一般有 $\rho=\lambda/(s\mu)$；

N：稳态系统任一时刻的状态(即系统中所有顾客数)；

U：任一顾客在稳态系统中的逗留时间；

Q：任一顾客在稳态系统中的等待时间.

其中，N,U,Q 都是随机变量.

对于损失制和混合制的排队系统，顾客在到达服务系统时，若系统容量已满，则顾客自行离去. 这就是说，到达的顾客不一定全部进入系统，为此引入 λ_e. 对于等待制的排队系统，有 $\lambda_e=\lambda$.

Little 公式 用 λ 表示有效平均到达率，即每单位时间内进入系统的平均顾客数(期望值). μ 表示单位时间内服务完毕离去的平均顾客数，因此，$1/\lambda$ 表示相邻两顾客到底的平均时间，$1/\mu$ 表示对每个顾客的平均服务时间，Little 给出了如下公式：

$$L_s=\lambda W_s, \qquad L_q=\lambda W_q$$

$$W_s=W_q+\frac{1}{\mu}, \qquad L_s=L_q+\frac{\lambda}{\mu}$$

与排队论有关的 LINGO 函数介绍：

(1) @peb(load,S). 该函数的返回值是当到达负荷为 load，服务系统中有 S 个服务台且允许排队时系统繁忙的概率，也就是顾客等待的概率.

(2) @pel(load,S). 该函数的返回值是当到达负荷为 load，服务系统中有 S 个服务台且不允许排队时系统损失概率，也就是顾客得不到服务离开的概率.

(3) @pfs(load,S,K). 该函数的返回值是当到达负荷为 load，顾客数为 K，平行服务台有 S 个时，有限源的泊松服务系统等待或返修顾客数的期望值.

排队系统中，由于顾客到达分布和服务时间分布是多种多样的，加之服务台数. 顾客源有限和无限，排队容量有限和无限等不同组合，就会有数不胜数的不同的排队模型，若对所有排队模型都进行分析与计算，不但十分繁杂而且也没有必要. 下面仅

分析几种常见排队系统模型：

1. 泊松输入-指数服务排队模型

泊松输入-负指数分布服务的排队系统的一般解决过程：

(1) 根据已知条件绘制状态转移速度图.

(2) 依据状态转移速度图写出各稳态概率之间的关系.

(3) 求出 P_0 及 P_n.

(4) 计算各项数量运行指标.

(5) 用系统运行指标构造目标函数，并对系统进行优化.

典型分布-泊松分布及其性质：负指数分布及其性质——泊松分布(平稳状态). $\lambda>0$为单位时间平均到达的顾客数.

$$P\{I=n\}=\frac{\lambda^n e^{-\lambda}}{n!},\quad n=0,1,2,\cdots$$

其中顾客等待概率为

$$P_{\text{wait}}=\text{@peb(load,s)}$$

其中 s 是服务台或服务员的个数，load 是系统到达负荷，即 load$=\lambda/\mu$.

顾客平均等待时间为

$$W_q=\frac{P_{\text{wait}}\cdot T}{S-\text{load}}$$

其中 $T/(S\text{-load})$是一个重要指标，可以看成一个“合理的长度间隔”. 此处需要注意的是：当 load 趋于 S 时，此值趋于无穷. 也就是说，系统负荷接近服务台的个数时，顾客平均等待时间将趋于无穷. 当 load$>S$ 时，该式无意义.

2. *M/M/1* 等待制排队模型

1) $M/M/1/\infty$负指数分布参数 λ,μ

例 5.7 某维修中心在周末只安排一名员工为顾客服务，新来维修的顾客到达后，若已有顾客正在接受服务，则需要排队等待. 假设来维修的顾客到达过程为泊松流，平均每小时 4 人，维修时间服从负指数分布，平均需要 6 min，试求该系统的主要数量指标.

记 $R=\lambda=4,T=1/\mu$(平均服务时间)，利用 LINGO 求解：

```
S=1;R=4;T=6/60;load=R*T;
Pwait=@peb(load,S);
W_q=Pwait*T/(S-load);L_q=R*W_q;
W_s=W_q+T;L_s=W_s*R;
```

2) $M/M/S/\infty:\lambda,\mu$

例 5.8 三个打字员，平均每个文件的打印时间为 10 min，而文件的到达率为每小时 15 件，试求该打印室的主要指标.

读者可以参考例 5.7LINGO 程序进行求解.

3) $M/M/1/1$ 损失制模型

顾客到达,若服务台被占用,则立即离开. 我们这里主要考察下面几个指标:

指标 1 系统损失概率:

$$\text{Plost}=@\text{pel}(\text{load},S)$$

指标 2 单位时间内平均进入系统的顾客数(R_c 或 λ_c):

$$R_c=\lambda_c=\lambda(1-\text{Plost})$$

指标 3 系统的相对通过能力 Q 与绝对通过能力 A:

$$Q=1-\text{Plost},\qquad A=\lambda c Q$$

指标 4 系统在单位时间内占用服务台的均值,即

$$L_s=\lambda c/\mu=R_c T\quad(\text{损失制中 } L_q=0)$$

指标 5 系统服务台的效率:

$$\eta=L_s/s$$

指标 6 顾客在系统内平均逗留时间(由于 $W_q=0$):

$$W_s=1/\mu=T$$

例 5.9 设某条电话线,平均每分钟有 0.6 次呼叫,若每次通话时间平均为 1.25 min,求系统相应的参数指标.

利用 LINGO 求解:

```
S=1;R=0.6;T=1.25;load=R*T;
Plost=@pel(load,S);
Q=1-Plost;R_e=Q*R;A=Q*R_e;
L_s=R_e*T;eta=L_s/S;
```

下面我们再来看一个例子:

例 5.10 某火车站售票处有三个窗口,同时售各车次的车票. 顾客到达服从泊松分布,平均每分钟到达 $\lambda=0.9$(人),服务时间服从负指数分布,平均服务率 $\mu=0.4$(人/min),记 $\rho=\lambda/(3\mu)$,分两种情况:①顾客排成一队,依次购票;②顾客在每个窗口排一队,不准串队. 求:

(1) 售票处空闲的概率;

(2) 平均等待时间和逗留时间;

(3) 队长和队列长.

分析 ①这是一个 $M/M/3/\infty/\infty$,

$$W_s=W_q+\frac{1}{\mu}=1.893+2.5=4.393\ (\text{min})$$

$$L_s=\lambda W_s=3.954$$

故售票处的空闲的概率为 0.0748,有一个窗口空闲的概率为 0.18934,有两个窗口空闲的概率为 0.1683. 平均等待时间 $W_q=1.893$ (min),平均逗留时间 $W=4.393$ (min).

队长 $L_s=3.954$(人),$L_q=1.704$(人).

② 现在问题为 $M/M/1/\infty/\infty$,三个系统并联,则

$$\lambda=0.3,\quad \mu=0.4,\quad \rho=\lambda/\mu=0.75,\quad P_0=1-\rho=0.25$$

三个服务台都有空的时候,

$$p_0^3=0.0156,\quad L_s=\frac{\rho}{1-\rho}=3\quad \lambda_e=\lambda=0.3$$

$$L_q=L_s-\frac{\lambda}{\mu}=2.25,\quad W_s=\frac{L_s}{\lambda}=10,\quad W_q=W_s-\frac{1}{\mu}=5.5$$

故售票处空闲的概率为 0.0156,有一个窗口空闲的概率为 0.25,有两个窗口空闲的概率为 0.0625. 平均等待时间为 $W_q=5.5$ min,平均逗留时间 $W_s=10$ min,队长 $L_s=3$,三个队共 $3+3+3=9$,队列长 $L_q=2.25$,共 6.75(人).

相比之下,排一队共享三个服务台效率更高.

例 5.11 某车站候车室在某段时间旅客到达服从泊松流分布,平均速度为 50 人/h,每位旅客在候车室内停留的时间服从负指数分布,平均停留时间为 0.5 h,问候车室内平均人数为多少?

将旅客停留在候车室视为服务,于是系统为 $M/M/\infty/\infty/\infty$,$\lambda=50$,$\mu=1/0.5=2$.

5.2.4 排队系统的优化目标与最优化问题

完全消除排队现象是不现实的,那将会造成服务人员和设施的严重浪费,但是设施的不足和低水平的服务,又将引起太多的等待,从而导致生产和社会性损失. 从经济角度考虑,排队系统的费用应该包含以下两个方面:一个是服务费用,它是服务水平的递增函数;另一个是顾客等待的机会损失(费用),它是服务水平的递减函数. 两者的总和呈一条 U 形曲线.

系统最优化的目标就是寻求上述合成费用曲线的最小点. 在这种意义下,排队系统的最优化问题通常分为两类:一类称为系统的静态最优设计,目的在于使设备达到最大效益,或者说,在保证一定服务质量指标的前提下,要求机构最为经济;另一类称为系统动态最优经营,是指一个给定排队系统,如何经营可使某个目标函数值达到最优.

归纳起来,排队系统常见的优化问题在于:①确定最优服务率 λ^*;②确定最佳服务台数量 s^*;③选择最为合适的服务规则;④确定上述几个量的最优组合.

有兴趣的读者可以查阅资料了解排队系统优化设计的基本思想. 本节仅给出一个简单的案例进行简要分析.

例 5.12 一车间内有 10 台相同的机器,每台机器运行时每小时能创造 60 元的利润,且平均每小时损坏 1 次,而一名修理工修复 1 台机器需要 15 min,以上时间均服从负指数分布,设一名修理工每小时工资为 90 元,问:

(1) 该车间应设置多少名修理工,使得总费用最少?

(2) 若要求损坏的机器等待修理的时间不超过 30 min,应设为多少名修理工?

这是 $M/M/s/K/K$ 模型,s 为未知量.

(1) $K=10$,设 L_s 为队长,则每小时工作的机器为 $K-L_{-s}$. 设置 s 名修理工. 所以,目标规划为 min $f=60L_s+90s$,LINGO 求解程序如下:

```
K=10;R=1;T=0.25;
L_s=@pfs(K*T*R,S,K);
R_e=R*(K-L_s);
P=(K-L_s)/K;
L_q=L_s-R_e*T;
W_s=L_s/R_e;
W_q=W_s-T;
Pwork=R_e/S*T;
min=60*L_S+90*s;
@gin(s);
```

求得结果 $s=2$,即设置两名修理工.

(2) $K=10$. 设 W_Q 为等待时间,则增加约束为 W_Q,设置 s 名修理工. 目标规划为 min $f=60L_s+90s$,利用 LINGO 求解的程序如下:

```
K=10;R=1;T=0.25;min=60*L_S+90*s;
L_s=@pfs(K*T*R,S,K);R_e=R*(K-L_s);P=(K-L_s)/K;L_q=L_s-R_e*T;
W_s=L_s/R_e;W_q=W_s-T;Pwork=R_e/S*T;W_Q<0.5;
    @gin(S);
```

求得结果 $s=2$,所以设置两名修理工.

评论 在使用排队论时,要注意该问题是属于排队论哪个模型,选择合适的模型解决问题,并且应注意顾客到达和服务时间的分布检验.

第 6 章　现代智能优化算法简介

6.1　遗传算法

生物的进化是一个奇妙的优化过程，它通过优胜劣汰，基因遗传、交叉、变异等规律产生适应环境变化的优良物种. 遗传算法是根据生物进化思想寻找全局解的一种概率优化算法.

遗传算法的基本思想是基于达尔文进化论和孟德尔的遗传学说的. 达尔文进化论最重要的是适者生存原理，它认为每一物种在发展中越来越适应环境. 物种每个个体的基本特征都由后代所继承，但后代又会产生一些异于父代的新变化. 在环境变化时，只有那些能适应环境的个体特征方能保留下来. 孟德尔遗传学说最重要的是基因遗传原理. 它认为遗传规律以密码方式存在细胞中，并以基因形式包含在染色体内. 每个基因有特殊的位置并控制某种特殊性质，所以，每个基因对环境具有某种适应性. 基因突变和基因杂交可能产生更适应于环境的后代. 经过存优去劣的自然淘汰，适应性高的基因得以保存下来.

遗传算法的重要应用领域是寻找全局解. 它能以概率 1 寻找到全局最优解. 遗传算法能够对搜索空间进行持续的搜索，因此遗传算法特别适合于在全局优化问题中应用. 遗传算法的特点：①直接对结构对象操作，不存在求导和函数连续性的限定；②遗传算法不是从单个定点，而是从一个群体的角度开始搜索；③具有内在的隐并行性和较好的全局寻优能力；④采用概率化寻优方法，能自动获取搜索过程中的有关知识并用于指导优化，自适应地调整搜索方向，不需要确定的规则；⑤鲁棒性强. 综上所述，可以知道遗传算法不仅能够处理连续的问题，而且能够较好地处理非连续、非线性、多目标等属于寻找全局解的问题，在算法上具有较好的收敛速率和适应性强等特点.

6.1.1　理论简介

1. 遗传算法的初识

引例　为 4 个连锁饭店寻找最好的经营决策，其中一个经营饭店的决策包括要做出以下三项决定：

(1) 价格. 汉堡包的价格应该定在 5 元还是 10 元？取 0 或 1 表示价格高、低.

(2) 饮料. 和汉堡包一起供应的应该是酒还是可乐？取 0 或 1 表示选取酒、可乐.

(3) 服务. 饭店应该提供慢的还是快的服务？取 0 或 1 表示选取慢、快服务.

目标：找到这三个决定的组合以产生最高的利润.

上述问题的二进制表示方案共有 8 种. 用遗传算法解这个问题的第一步就是选取一个适当的表示方案. 群体规模 $N=4$，方案表示见表 6.1，适应值表示适应程度，同时代表收益值，表示见表 6.2.

表 6.1　饭店问题的表示方案

饭店编号	价格	饮料	速度	二进制表示
1	高	可乐	快	011
2	高	酒	快	001
3	低	可乐	慢	110
4	高	可乐	慢	010

表 6.2　初始群体中经营决策的适应值

第 0 代		
i	串 x_i	适应值 $f(x_i)$
1	011	3
2	001	1
3	110	6
4	010	2
总和		12
最小值		1.0
平均值		3.0
最大值		6.0

注：适应值表示目标函数的取值.

一个简单的遗传算法由复制、杂交、变异三个算子组成.

复制算子　采用轮盘选择. 根据表 6.3 中 $f(x_i)/\sum f(x_i)$ 值进行复制基因遗传过程（显然，比值越大的串被复制的可能性越大，如串 110 被复制了两次）.

表 6.3　使用复制算子后产生的交配池

第 0 代				交配池	
i	串 x_i	适应值 $f(x_i)$	$f(x_i)/\sum f(x_i)$	新串	新适应值 $f(x_i)$
1	011	3	0.25	011	3
2	001	1	0.08	110	6
3	110	6	0.50	110	6
4	010	2	0.17	010	2
总和 $\sum f(x_i)$		12			17

续表

第 0 代			交配池	
最小值	1.0			2.00
平均值	3.0			4.25
最大值	6.0			6.00

杂交算子　采用一点杂交作用过程：①产生一个落在$[1,l-1]$的随机整数 i，即杂交位点．②配对的两个串相互对应的交换从 $i+1$ 到 l 的位段．例如，从交配池中选择编号为 1 和 2 的串进行配对，且杂交点选在 2(随机数 $i=2$，用分隔符"|"表示)，杂交算子作用的结果为

$$
\begin{array}{ll}
01|1 & 010 \\
11|0 & 111
\end{array}
$$

对交配池中指定百分比的个体应用杂交算子，假设杂交概率 $p_c=50\%$，则交配池中 50%个体进行杂交运算．交配详细过程见表 6.4.

表 6.4　使用复制和杂交算子的作用结果

	第 0 代			交配池		第 1 代		
i	串 x_i	$f(x_i)$	$f(x_i)/\sum f(x_i)$	串	$f(x_i)$	杂交点	x_i	$f(x_i)$
1	011	3	0.25	011	3	2	010	2
2	001	1	0.08	110	6	2	111	7
3	110	6	0.50	110	6	—	110	6
4	010	2	0.17	010	2	—	010	2
总和 $\sum f(x_i)$		12			17			17
最小值		1			2			2
平均值		3.00			4.25			4.25
最大值		6			6			7

遗传算法利用复制和杂交算子可以产生具有更高平均适应值和更好个体的群体.

变异算子　以一个很小的概率 p_m 随机改变染色体串上的某个位．对于二进制串，就是将相应位上的 0 变为 1 或将 1 变为 0．例如，选交配池中编号为 4 的串进行变异，且变异点在 2，则

$$010 \longrightarrow 000$$

变异算子相对而言是次要算子，但在恢复群体中失去的多样性方面具有潜在的作用.

上述遗传算法描述了从第 0 代产生第 1 代的过程，然后遗传算法迭代地执行这个过程，直到满足某个停止准则．在每一代中，算法首先计算群体中每个个体的适应

值，然后利用适应值信息，遗传算法分别以概率 p_c 和 p_m 执行杂交和变异操作，从而产生新的群体.

2. 基本遗传算法的描述

应用遗传算法求解问题需完成 4 个主要步骤：①确定表示方案；②确定适应值度量；③设计遗传算子；④确定控制算法的参数和变量.

1）染色体编码方法

基本遗传算法使用固定长度的二进制符号串来表示群体中的个体，其等位基因由二值符号集{0,1}组成. 初始群体中每个个体的基因值用均匀分布的随机数来生成. 例如，x：100111001000101101 就可表示一个个体，该个体的染色体长度是 $L=18$.

2）个体适应度评价——目标函数

基本遗传算法按与个体适应度成正比的概率来决定当前群体中每个个体遗传到下一代群体中的机会多少. 为正确计算这个概率，这里要求所有个体的适应度必须为正数或零. 这样，根据不同种类的问题，必须预先确定好由目标函数值到个体适应度之间的转换规则，特别是要预先确定好当目标函数值为负数时的处理方法.

3）遗传算子

基本遗传算法使用下述三种遗传算子：

(1) 复制(选择)运算：使用比例选择算子.

(2) 杂交(交叉)运算：使用单点杂交算子.

(3) 变异运算：使用基本位变异算子.

4）基本遗传算法的运行参数

基本遗传算法有下述 4 个运行参数需要提前设定.

N：群体大小，一般取为 20～100；

T：终止进化代数，一般取为 100～500；

p_c：交叉概率，一般取为 0.4～0.99；

p_m：变异概率，一般取为 0.0001～0.1.

说明　这 4 个运行参数对遗传算法的求解结果和求解速率都有一定的影响，但目前尚无合理选择它们的理论依据. 在遗传算法的实际应用中，往往需要经过多次试算后才能确定出这些参数合理的取值大小或取值范围.

3. 基本遗传算法的实现

根据上面对基本遗传算法构成要素的分析和描述，我们可以很方便地用计算机语言来实现这个基本遗传算法. 现对具体实现过程中的问题作以下说明：

1）编码与解码

(1) 编码. 假设某一变量(个体表示变量)的取值范围是$[u_{min}, u_{max}]$，用长度为 l 位的二进制 0,1 编码符号串来表示该参数，则它总共能够产生 2^l 种不同的编码，参数编码时的对应关系如下：

$$00000000\cdots00000000=0 \qquad u_{\min}$$

$$00000000\cdots00000001=1 \qquad u_{\min}+\delta$$

……

$$11111111\cdots11111111=2^l-1 \qquad u_{\max}$$

其中,δ 为二进制编码的编码精度,其公式为

$$\delta=\frac{u_{\max}-u_{\min}}{2^l-1}$$

(2) 解码.假设某一个体的编码是 $x:b_lb_{l-1}b_{l-2}\cdots b_2b_1$,其中 $b_i=0$ 或 1,则对应的解码公式为

$$x=u_{\min}+\left(\sum_{i=1}^{l}b_i\cdot 2^{i-1}\right)\cdot\frac{u_{\max}-u_{\min}}{2^l-1}$$

例如,设 $-3.0\leqslant x\leqslant 12.1$,精度要求 $\delta=1/10000$,由公式,得

$$2\cdot\frac{u_{\max}-u_{\min}}{\delta}+1=\frac{12.1+3.0}{1/10000}+1$$

即
$$2^{17}<151001<2^{18}$$

x 需要 18 位{0,1} 符号表示.例如,010001001011010000.解码:

$$x=u_{\min}+\left(\sum_{i=1}^{l}b_i\cdot 2^{i-1}\right)\cdot\frac{u_{\max}-u_{\min}}{2^l-1}$$
$$=-3.0+70352\times(12.1+3)/(2^{18}-1)=1.052426$$

2) 个体适应度评价

如前所述,要求所有个体的适应度必须为正数或零,不能是负数.

(1) 当优化目标是求函数最大值,并且目标函数总取正值时,可以直接设定个体的适应度 $F(x)$ 就等于相应的目标函数值 $f(x)$,即 $F(x)=f(x)$.

(2) 对于求目标函数最小值的优化问题,理论上只需简单地对其增加一个负号就可将其转化为求目标函数最大值的优化问题,即 $\min f(x)=\max\{-f(x)\}$.

但实际优化问题中的目标函数值有正也有负,优化目标有求函数最大值,也有求函数最小值,显然上面两式保证不了所有情况下个体的适应度都是非负数这个要求.

基本遗传算法一般采用下面两种方法之一将目标函数值 $f(x)$ 变换为个体的适应度 $F(x)$.

方法一 对于求目标函数最大值的优化问题,变换方法为

$$F(x)=\begin{cases}f(x)+c_{\min}, & f(x)+c_{\min}>0\\ 0, & f(x)+c_{\min}\leqslant 0\end{cases}$$

其中,$c_{\min}$ 为一个适当的相对比较小的数,它可用下面方法之一来选取:

① 预先指定的一个较小的数;

② 进化到当前代为止的最小目标函数值的绝对值;

③ 当前代或最近几代群体中的最小目标函数值.

方法二　对于求目标函数最小值的优化问题,变换方法为

$$F(x)=\begin{cases}c_{\max}-f(x), & f(x)<c_{\max}\\ 0, & f(x)\geqslant c_{\max}\end{cases}$$

其中,$c_{\max}$是一个适当的相对比较大的数,它可用下面几种方法求得:

① 预先指定的一个较大的数;

② 进化到当前代为止的最大目标函数值;

③ 当前代或最近几代群体中的最大目标函数值.

3) 复制(选择)算子

(1) 复制算子或称为选择算子的作用:从当前代群体中选择出一些比较优良的个体,并将其复制到下一代群体中.

(2) 最常用和最基本的选择算子:比例选择算子.

(3) 比例选择算子:指个体被选中并遗传到下一代群体中的概率与该个体的适应度大小成正比.

(4) 执行比例选择的手段是轮盘选择.

轮盘法的基本精髓是,个体被选中的概率取决于个体的相对适应度:

$$p_i = f_i/\sum f_i,\quad i=1,2,\cdots,M$$

其中,p_i 表示个体 i 被选中的概率,f_i 表示个体 i 的适应度,$\sum f_i$ 表示群体的累加适应度.

显然,个体适应度愈高,被选中的概率愈大. 但是,适应度小的个体也有可能被选中,以便增加下一代群体的多样性.

轮盘选择的原理:图 6.1 中指针固定不动,外圈的圆环可以自由转动,圆环上的刻度代表各个个体的适应度. 当圆环旋转若干圈后停止,指针指定的位置便是被选中的个体. 从统计意义讲,适应度大的个体,其刻度长,被选中的可能性大;反之,适应度小的个体被选中的可能性小,但有时也会被“破格”选中.

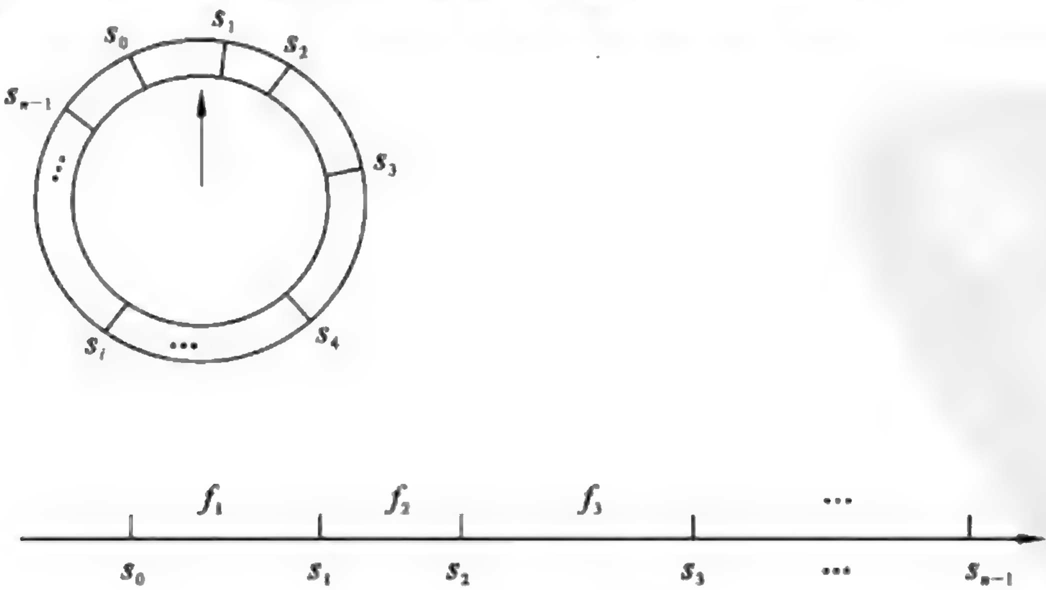


图 6.1　轮盘示意图

4）单点交叉算子

（1）交叉算子作用：通过交叉，子代的基因值不同于父代. 交换是遗传算法产生新个体的主要手段. 正是有了交换操作，群体的性态才多种多样.

（2）最常用和最基本——单点交叉算子.

（3）单点交叉算子的具体计算过程如下：

① 对群体中的个体进行两两随机配对. 若群体大小为 M，则共有$[M/2]$对相互配对的个体组.

② 每一对相互配对的个体，随机设置某一基因位之后的位置为交叉点. 若染色体的长度为 l，则共有 $l-1$ 个可能的交叉点位置.

③ 对每一对相互配对的个体，依设定的交叉概率 p_c 在其交叉点处相互交换两个个体的部分染色体，从而产生出两个新的个体. 单点交叉运算的示例如下所示：

A:10110111　00　　　　A':10110111　11

B:00011100　11　　　　B':00011100　00

交叉概率：

$$p_c=\frac{N_c}{N}$$

其中，N 表示群体中个体的数目，N_c 表示群体中被交换个体的数目.

交叉操作示例：交叉的个体是随机确定的，见表 6.5. 某群体有 n 个个体，每个个体含 8 个等位基因. 针对每个个体产生一个[0,1]区间的均匀随机数. 假设交叉概率 $p_c=0.6$，则随机数小于 0.6 的对应个体与其随机确定的另一个个体交叉，交叉点随机确定.

表 6.5　交叉操作示例

个体编号	个体	随机数	交叉操作	新个体
1	11011000	0.728	11011000	
2	10101011	0.589	101010　11	101010　01
3	00101100	0.678	00101100	
4	10001101	0.301	100011　01	100011　11
⋮	⋮	⋮	⋮	⋮

5）基本位变异算子

基本位变异算子是最简单和最基本的变异操作算子. 对于基本遗传算法中用二进制编码符号串所表示的个体，若需要进行变异操作的某一基因位上的原有基因值为 0，则变异操作将该基因值变为 1，反之，若原有基因值为 1，则变异操作将其变为 0. 基本位变异因子的具体执行过程如下：

（1）对个体的每一个基因位，依变异概率 p_m 指定其为变异点.

（2）对每一个指定的变异点，对其基因值做取反运算或用其他等位基因值来代

替，从而产生出一个新的个体.

基本位变异运算的示例如下所示：

$$A:\ 1010\ \boxed{1}\ 01010 \xrightarrow{\text{基本位变异}} A':\ 1010\ \boxed{0}\ 01010$$

变异概率：变异是针对个体的某一个或某一些基因位上的基因值执行的，因此变异概率 p_m 也是针对基因而言，即

$$p_m=\frac{B}{Nl}$$

其中，B 表示每代中变异的基因数目，N 表示每代中群体拥有的个体数目，i 表示个体中基因串长度.

变异操作示例：变异字符的位置是随机确定的，见表 6.6. 某群体有 3 个个体，每个个体含 4 个基因. 针对每个个体的每个基因产生一个[0,1]区间内具有 3 位有效数字的均匀随机数. 假设变异概率 $p_m=0.01$，则随机数小于 0.01 的对应基因值产生变异. 表 6.6 中 3 号个体的第 4 位的随机数为 0.001，小于 0.01，该基因产生变异，使 3 号个体由 0010 变为 0011. 其余基因的随机数均大于 0.01，不产生变异.

表 6.6　变异操作示例

个体编号	1,0 个体	随机数				新个体
1	1010	0.801	0.102	0.266	0.373	
2	1100	0.120	0.796	0.105	0.840	
3	0010	0.760	0.473	0.894	0.001	0011

6.1.2　案例分析

1. 例子表述

例 6.1　计算下列函数的全局最大值：

$$\max f(x)=x+10\sin 5x+7\cos 4x$$
$$\text{s.t.}\quad 0\leqslant x\leqslant 9$$

2. 分析与建模

分析　如图 6.2 所示，该函数有多个局部极大点.

第一步　确定决策变量及其约束条件.

$$\text{s.t.}\ \ 0\leqslant x\leqslant 9$$

第二步　建立优化模型.

$$\max f(x)=x+10\sin 5x+7\cos 4x$$

第三步　确定编码方法. 用长度为 22 位的二进制编码串来表示决策变量 x. 22 位二进制编码串可以表示从 0 到 4194303 之间的 4194304 个不同的数，故将 x 的定义域

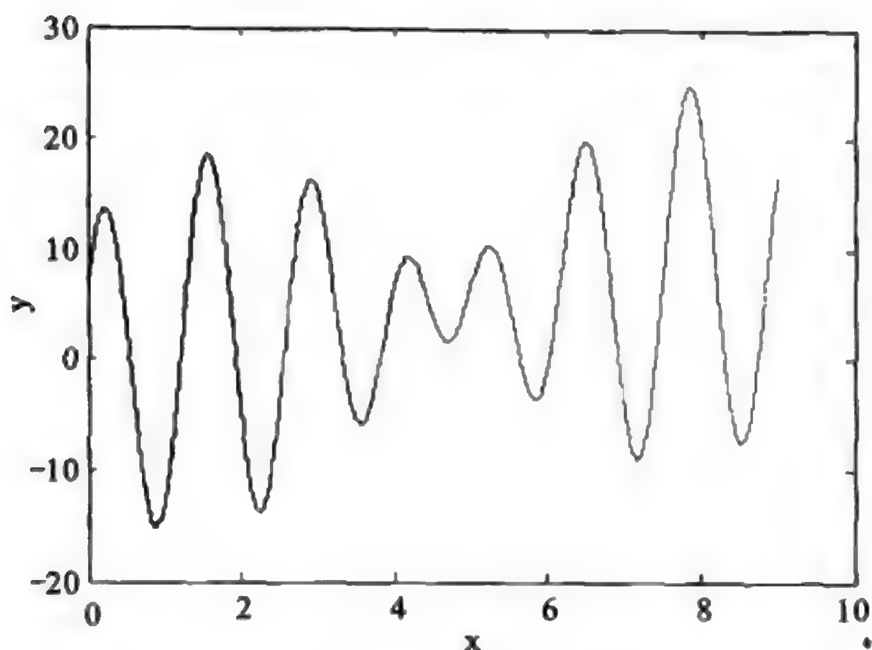


图 6.2 $x+10\sin 5x+7\cos 4x$ 函数图形构造过程

离散化为 4194304 个均等的区间，包括两个端点在内共有 4194304 个不同的离散点. 从离散点 0 到离散点 9，依次让它们分别对应于从 0000000000000000000000(0) 到 1111111111111111111111(41943043)之间的二进制编码. 这就构成了这个函数优化问题的染色体编码方法. 例如，X:0000000000001101110001 就表示一个个体的基因型.

第四步 确定解码方法. 解码时将 22 位长的二进制编码转换为对应的十进制整数代码，记为 y. 依据前述个体编码方法相对定义域的离散化方法可知，将代码 y 转换为变量 x 的解码公式为

$$x=9\cdot\frac{y}{4194303}$$

例如，对前述个体 X:00000000001101110001，它由下列代码所组成：$y=881$. 经解码处理后，得：$x=0.001890421$.

第五步 确定个体评价方法. 由于目标函数的取值可正可负，并且优化目标是求函数的最大值，故将个体的适应度取为每次迭代的最小值的绝对值加上目标函数值，即有

$$F(x)=|\min\{f(x)\}|+f(x)$$

第六步 设计遗传算子. 选择运算使用比例选择算子，交叉运算使用单点交叉算子，变异运算使用基本位变异算子.

第七步 确定遗传算法的运行参数. 对于本例，设定基本遗传算法的运行参数如下：群体大小 $N=50$；终止代数 ger=100；交叉概率 $p_c=0.65$；变异概率 $p_m=0.05$.

3. 编程实现

由于程序源码过长，读者请登陆华中数学建模网 http://www.shumo.cn/大学数学实验栏目下载学习.

4. 结果分析

图 6.3 为其进化过程示例及运行结果. 图中两条曲线分别为各代群体中个体适

应度的最大值和平均值.

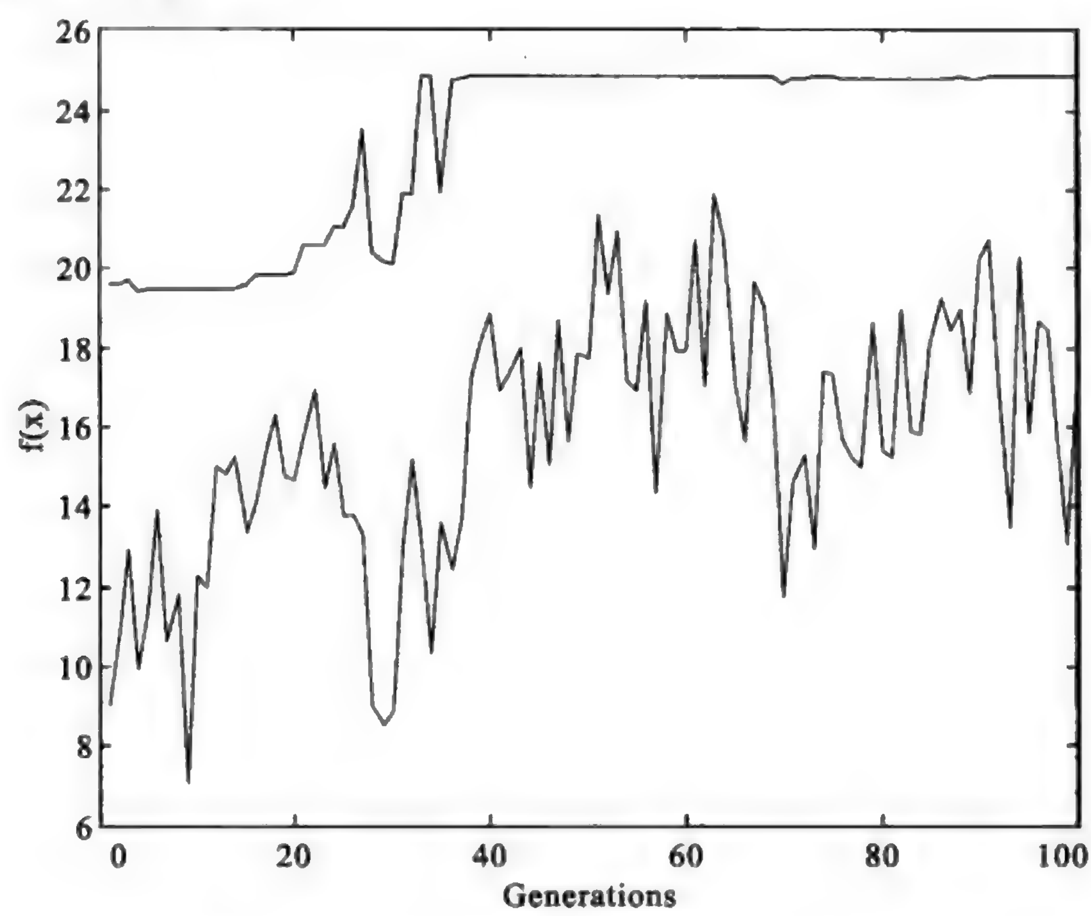


图 6.3　个体适应度的最大值和平均值

图 6.4 分别为初始群体和进化结束后群体中个体的分布情况.

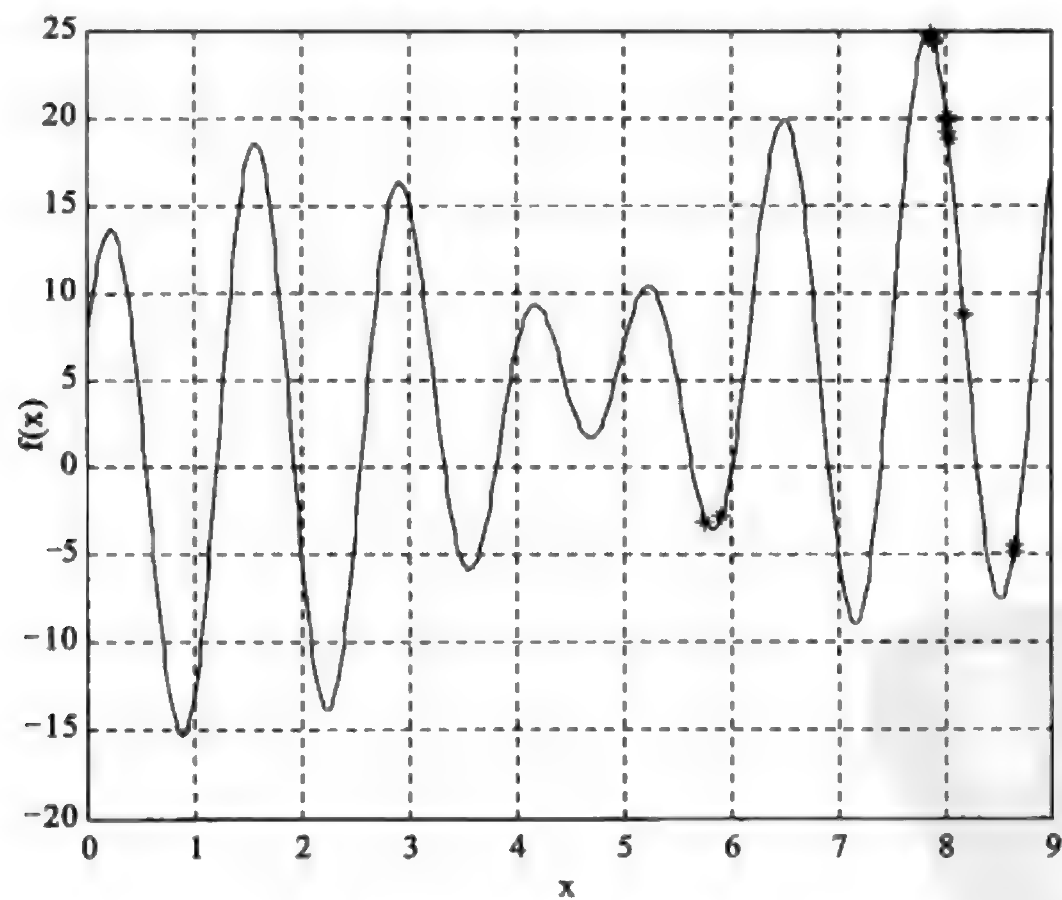


图 6.4　染色体的最终位置

由该组图我们可以看出,随着进化过程的进行,群体中适应度较低的一些个体被逐渐淘汰掉,而适应度较高的一些个体会越来越多.并且它们都集中在所求问题的最

优点附近，从而最终可搜索到问题的最优解.

6.1.3 评论、体会与展望

遗传算法是一种概率化算法，主要思想是利用“生存法则”优胜劣汰，使用交叉算子和变异算子打破局部最优(稳定点)，从而获得了全局最优. 算法的核心步骤是染色体的编码、解码和 $F(x)$ 的选取. 虽然说遗传算法是以概率 1 趋于最优解，但是算法所需要的时间是有限的. 所以算法所需要的时间会在一定程度上取决于初值的选取，这也是我们能够改进算法的地方之一. 我们这里仅仅给出了例 6.1 的求解，更进一步说，这里仅仅是给出了一种二进制的编码方法，其实可以使用十进制等编码，这主要是取决于要解决的问题. 所以这是我们可以进一步学习的地方. 其次，正如上面所说，算法会在某种程度上决定与初值的选取，我们可以通过选取合适的初值来缩短程序运行所需要的时间. 具体的改进方案这里不予给出. 最后，算法具有很强的兼容性和扩展性，可以和很多的其他优化算法相结合，这样可以克服自身收敛速度减慢的缺点，同时可以保持全局搜索能力.

6.2 蚁群算法

20 世纪 90 年代意大利学者 M. Dorigo, V. Maniezzo, A. Colorni 等从生物进化的机制中受到启发，通过模拟自然界蚂蚁搜索路径的行为，提出来一种新型的模拟进化算法——蚁群算法，是群智能理论研究领域的一种主要算法. 用该方法求解 TSP 问题、分配问题、job-shop 调度问题，取得了较好的试验结果. 虽然研究时间不长，但是现在的研究显示出，蚁群算法在求解复杂优化问题(特别是离散优化问题)方面有一定优势，表明它是一种有发展前景的算法. 这种方法能够被用于解决大多数优化问题或者能够转化为优化求解的问题. 现在其应用领域已扩展到多目标优化、数据分类、数据聚类、模式识别、生物系统建模、流程规划、信号处理、机器人控制、决策支持以及仿真和系统辨识等方面，群智能理论和方法为解决这类应用问题提供了新的途径.

6.2.1 理论简介

1. 蚁群算法原理

蚁群算法是对自然界蚂蚁的寻径方式进行模拟而得出的一种仿生算法. 蚂蚁在运动过程中，能够在它所经过的路径上留下一种称为外激素(pheromone)的物质进行信息传递，而且蚂蚁在运动过程中能够感知这种物质，并以此指导自己的运动方向，因此由大量蚂蚁组成的蚁群集体行为便表现出一种信息正反馈现象：某一路径上走过的蚂蚁越多，则后来者选择该路径的概率就越大.

为了说明蚁群算法的原理，先简要介绍一下蚂蚁搜寻食物的具体过程. 在蚁群

寻找食物时，它们总能找到一条从食物到巢穴之间的最优路径. 这是因为蚂蚁在寻找路径时会在路径上释放出一种特殊的信息素. 当它们碰到一个还没有走过的路口时，就随机地挑选一条路径前行. 与此同时释放出与路径长度有关的信息素. 路径越长，释放的激素浓度越低. 当后来的蚂蚁再次碰到这个路口的时候，选择激素浓度较高路径概率就会相对较大. 这样形成一个正反馈. 最优路径上的激素浓度越来越大. 而其他的路径上激素浓度却会随着时间的流逝而消减. 最终整个蚁群会找出最优路径.

2. 简化的蚂蚁寻食过程

如图 6.5 所示，蚂蚁从 A 点出发，速度相同，食物在 D 点，可能随机选择路线 ABD 或 ACD. 假设初始时每条分配路线一只蚂蚁，每个时间单位行走一步，本图为经过 9 个时间单位时的情形：走 ABD 的蚂蚁到达终点，而走 ACD 的蚂蚁刚好走到 C 点，为一半路程.

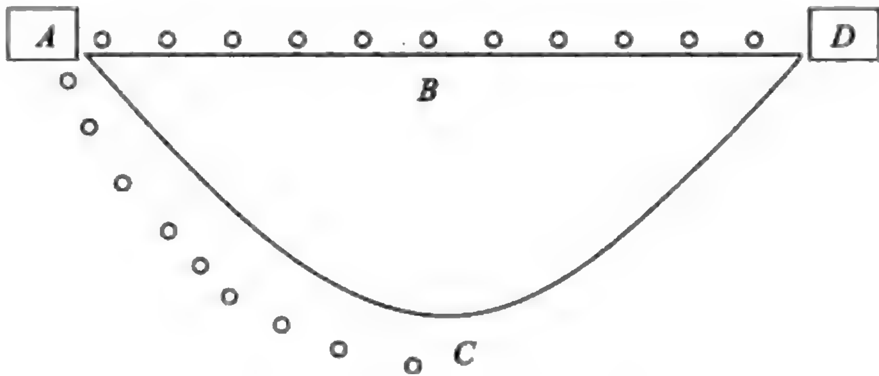


图 6.5 蚂蚁寻食示意图(一)

图 6.6 为从开始算起，经过 18 个时间单位时的情形：走 ABD 的蚂蚁到达终点后得到食物又返回了起点 A，而走 ACD 的蚂蚁刚好走到 D 点. 假设蚂蚁每经过一处所留下的信息素为一个单位，则经过 36 个时间单位后，所有开始一起出发的蚂蚁都经过不同路径从 D 点取得了食物，此时 ABD 的路线往返了 2 趟，每一处的信息素为 4 个单位，而 ACD 的路线往返了一趟，每一处的信息素为 2 个单位，其比值为 2∶1. 寻找食物的过程继续进行，则按信息素的指导，蚁群在 ABD 路线上增派一只蚂蚁(共 2 只)，而 ACD 路线上仍然为一只蚂蚁. 再经过 36 个时间单位后，两条线路上的信息素单位积累为 12 和 4，比值为 3∶1. 若按以上规则继续，蚁群在 ABD 路线上再增派一只蚂蚁(共 3 只)，而 ACD 路线上仍然为一只蚂蚁. 再经过 36 个时间单位后，两条线路上的信息素单位积累为 24 和 6，比值为 4∶1. 若继续进行，则按信息素的指导，最终所有的蚂蚁会放弃 ACD 路线，而都选择 ABD 路线. 这也就是前面所提到的正反馈效应.

3. 自然蚁群与人工蚁群算法

基于以上蚁群寻找食物时的最优路径选择问题，可以构造人工蚁群，来解决最优化问题. 人工蚁群中将具有简单功能的工作单元视为蚂蚁. 两者的相似之处在于都是优先选择信息素浓度大的路径，较短路径的信息素浓度高，所以能够最终被所有蚂

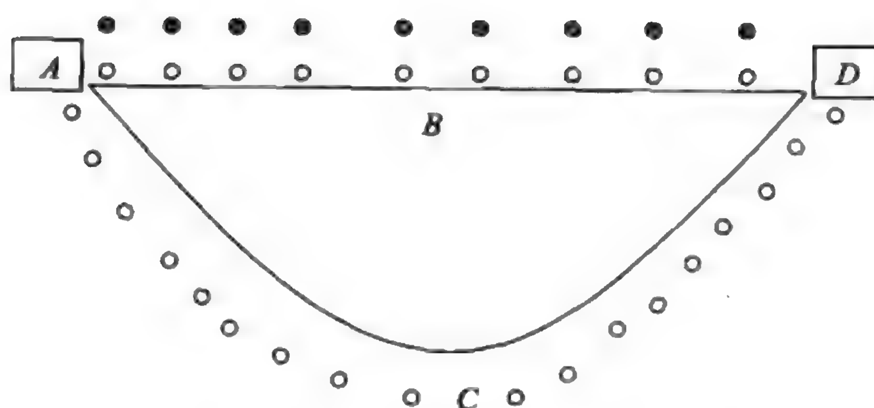


图 6.6　蚂蚁寻食示意图(二)

蚁选择,也就是最终的优化结果;两者的区别在于人工蚁群有一定的记忆能力,能够记忆已经访问过的结点.同时,人工蚁群再选择下一条路径的时候是按一定算法规律有意识地寻找最短路径,而不是盲目的.

4. 实数编码的小生境蚁群算法(NACA)(伪码实现)

NACA 的基本思路:随机产生 N 个蚂蚁的初始群体,使蚂蚁随机分布在函数的可行域上,根据优化函数计算每只蚂蚁的初始信息素,信息素正比于函数值,根据每只蚂蚁的当前信息素和全局最优信息素求出蚂蚁的转移概率,根据转移概率更新每只蚂蚁的位置,新位置限制在函数可行域内,蚂蚁移动到新位置后就立即更新自己的信息素.

NACA 伪代码描述如下:

```
P0=0.7;  %P0为全局转移概率
P=0.3;  %P为信息素蒸发系数
For each ant
  X[i]=(start+(end-start)*rand(1));
%随机产生蚂蚁的初始位置(限制在可行域内[start,end])
  T[i]=k*f(X[i]);  %计算每只蚂蚁的初始信息素(正比于函数值),k为比例常数
End
DO  %循环迭代
  T_Best=max(T);  %求出最大信息素
  For each ant
    Prob[i]=(T_Best- T[i])/T_Best;  %求每只蚂蚁的下一步转移概率
    End
    For each ant
      If Prob[i]<P0
      Temp=X[i]+min_step*(rand(1)-0.5);  %局部搜索
    For each ant
      If Prob[i]<P0
      Temp=X[i]+min_step*(rand(1)-0.5);  %局部搜索
    Else
```

```
      Temp=X[i]+max_step*(rand(1)-0.5);   %全局搜索
    End
    把 Temp 限制在可行域内[start,end];
    把 Temp 限制在可行域内[start,end];
    If f(Temp)>f(X[i])   %目标函数值比较
       X[i]=Temp;   %更新蚂蚁的位置成功
    End
  End
  For each ant
    T[i]=(1-P)*T[i]+k*f(X[i]);   %更新每只蚂蚁的信息素
  End
While(设定的最大迭代次数);
```

6.2.2 案例分析

1. 例子的表述

例 6.2 用蚁群算法求下面加权有向图(图 6.7)中从 A 到 G 的最短路.

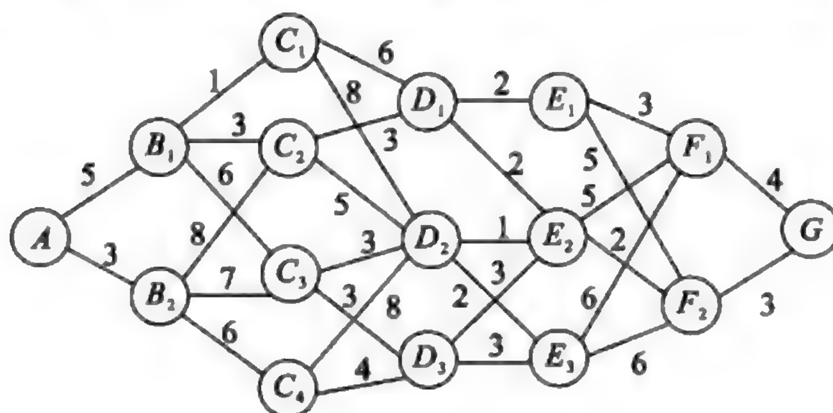


图 6.7 加权有向图

2. 分析与建模

分析 将点 1～16 是否在路径上分别取值为 0 或 1,这样就形成了 16 位的 0,1 序列,从而计算这条路径的距离. 将距离作为信息素的一个映射变量,由于这里要求最短路,所以可以使用倒数或者相对距离作为信息素浓度. 这样就可以获得每只蚂蚁的转移概率,若转移概率大于全局转移因子,则进行全局转移;否则以一定得步长进行转移. 这样就可以逐步向全局最优解靠近.

第一步 初始化 N 只蚂蚁. 实际上就是 N 条道路,并计算当前的蚂蚁位置、作图.

第二步 初始化运行参数,开始迭代.

第三步 在迭代步数范围之内,计算转移概率,如果小于全局转移概率就进行小范围的搜索,否则进行大范围的搜索.

第四步 更新信息素,记录状态,准备进行下一次迭代.

第五步 转第三步.

第六步 输出结果.

3. 编程实现

源代码程序详见华中数学建模网 http://www.shumo.cn/大学数学实验栏目.

4. 结果分析

运行结果如下：

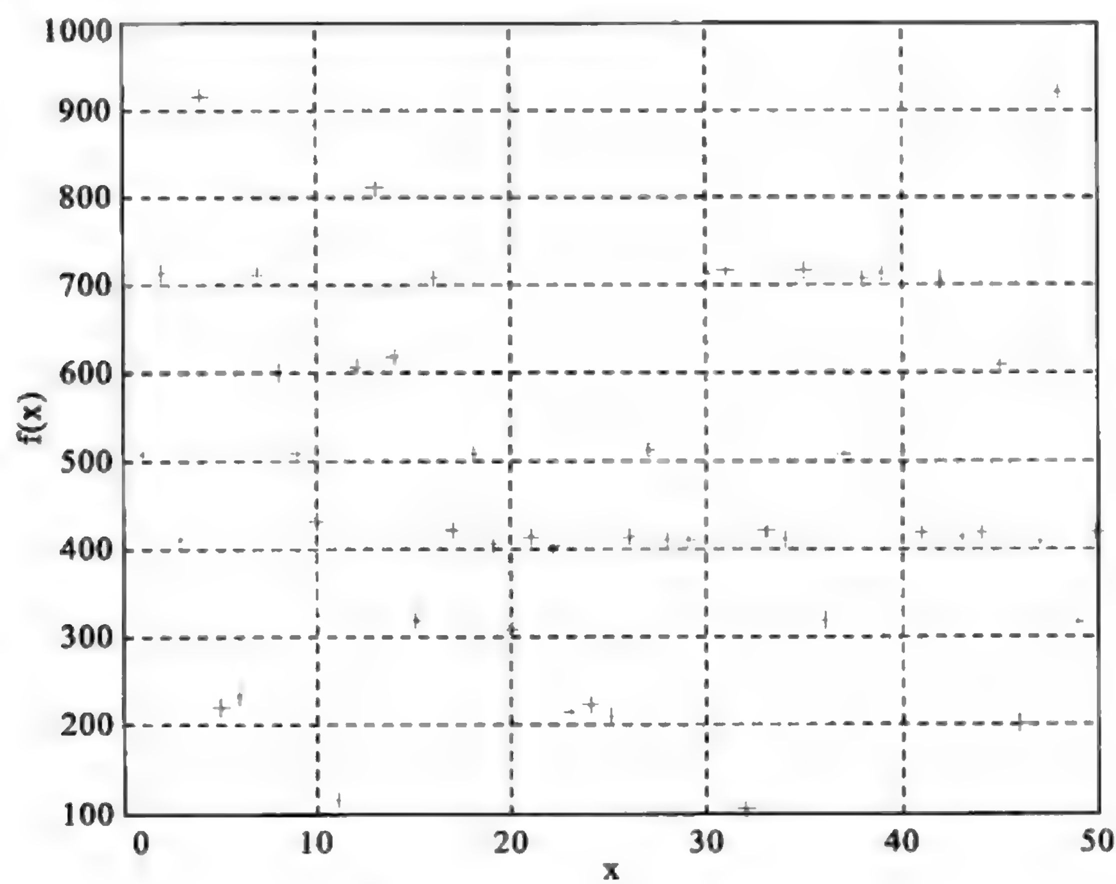


图 6.8　蚂蚁的初始位置

图 6.8 表示了 50 只蚂蚁的初始状态的无序分布情况.

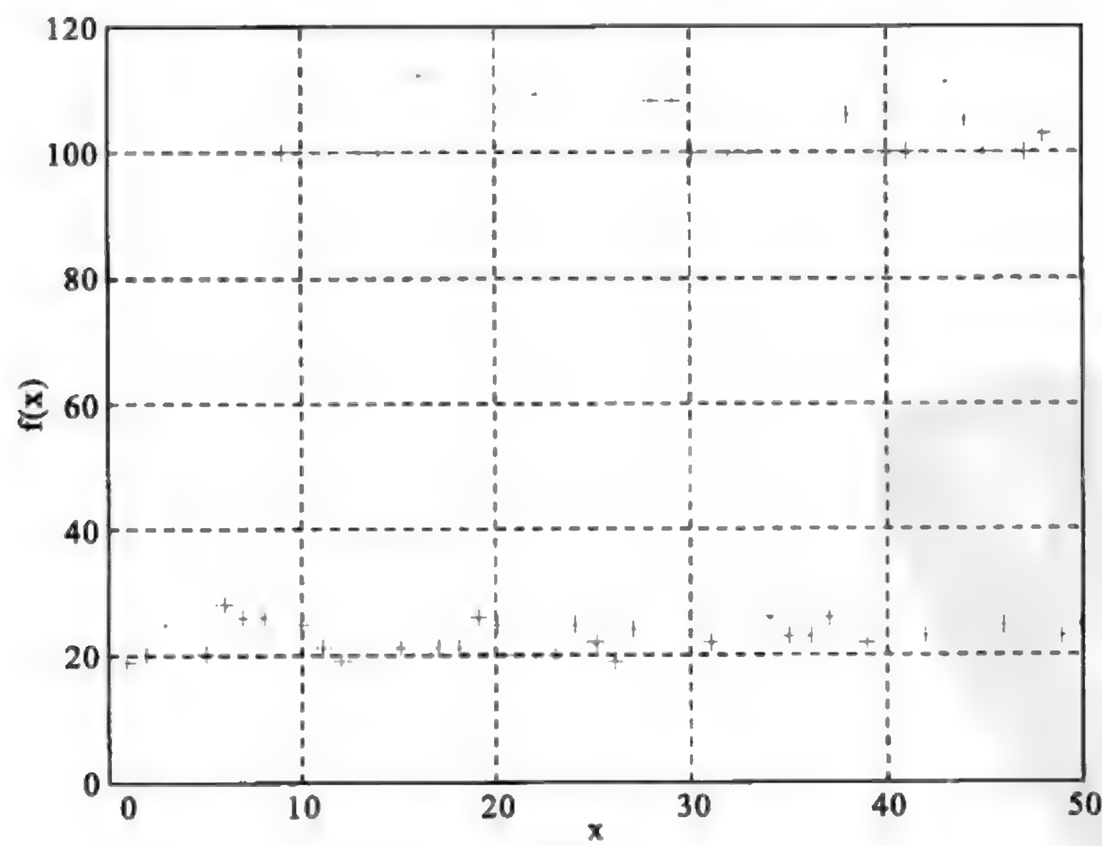


图 6.9　蚂蚁的最终位置

图 6.9 表示蚂蚁移动后的位置,优化后蚂蚁实现了两极分化,这样我们就得到了最优解.

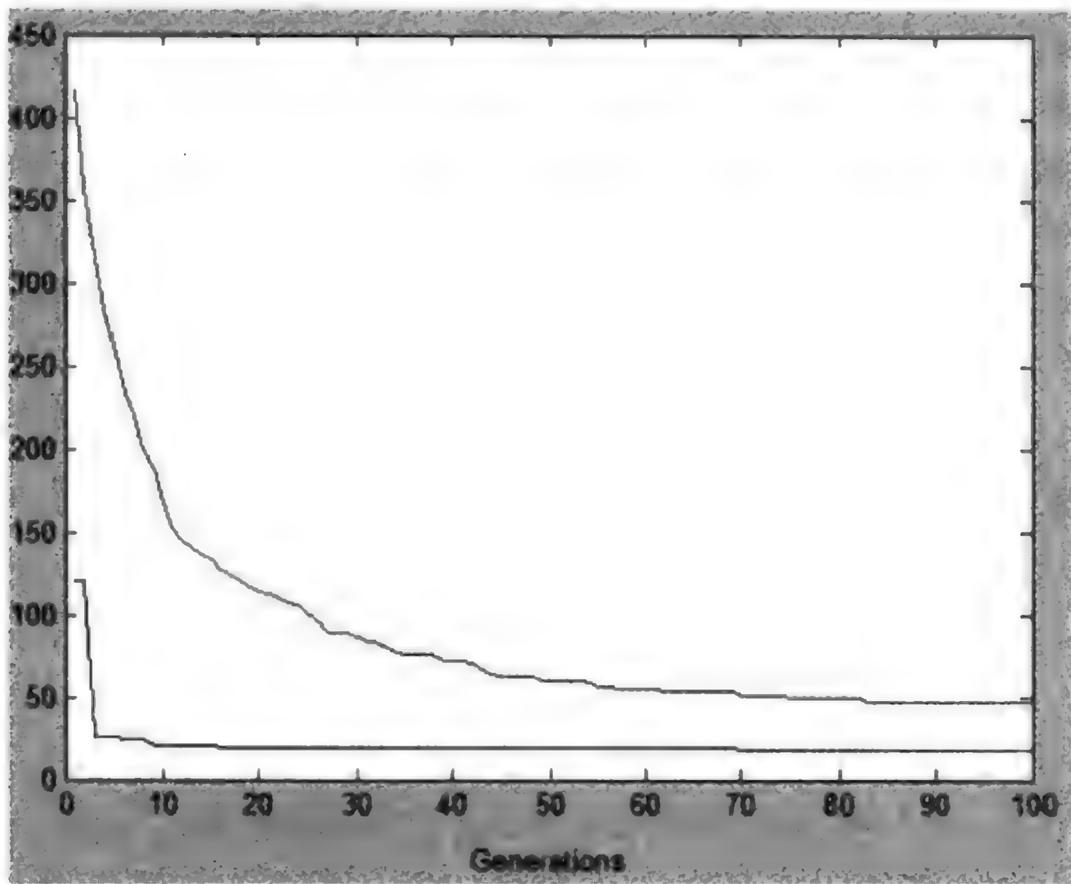


图 6.10　信息素浓度的平均值和最优值

图 6.10 分别是平均和最优的变化曲线,从中可以知道,算法收敛速度很快,效果较好.

最短路是 1→2→5→8→12→15→16. 距离是 18. 染色体是:1100100100010011. 运行时间是 1.0648(s).

6.2.3　评论、体会与展望

蚁群算法的主要思想是根据信息素的浓度高低决定蚂蚁的进化速率. 该算法具有较强的局部搜索能力,收敛速度较快,便于对离散的问题进行求解等优点,但是算法有些时候得不到全局最优解,只能得到局部解,这就迫使我们增加更多的条件,对于连续的问题不得不对连续的空间进行分割. 算法的核心步骤是:①蚂蚁的表示;②变量和信息素之间的映射;③前进方向的确定. 蚁群算法具有较强的局部搜索能力和收敛速度较快的特点,要是能够与其他的优化算法相结合将会产生一些较好的优化算法.

6.3　其他优化算法简介

6.3.1　贪婪算法

1. 贪婪算法(greedy method)说明

对于一个优化决策问题,人们通常最直接的反应是看能不能将这个问题的全部

状态都列举出来，再进行比较，这是一种穷举的思想，但当一个问题的状态空间很大时，穷举法计算量可能会太大. 于是现实中当人们面对一个这样的问题时，可能会采取目前看来最接近解状态的选择方案. 这就是一种贪婪算法的思想.

贪婪算法是一种不追求最优解，只希望得到较为满意解的方法. 有些问题，只有通过系统地、彻底地搜索所有可能解才能得到最优解，从而必须耗费大量时间，使得求出最优解的代价太高，有些复杂问题甚至于在现实条件下都不可能完成穷举. 这时，如果可以快速地求得一个与最优解相差不多的近似解(或次优解)就可以满足要求(满意解). 此时我们就可以考虑选用贪婪算法. 贪婪算法常以当前情况为基础作最优选择，而不考虑各种可能的整体情况. 贪婪算法采用逐步构造最优解的方法. 一般贪婪算法将构造可行解的工作分阶段来完成，在每个阶段，选择那些在一定的标准下是局部最优的方案，期望各阶段的局部最优的选择带来整体最优. 决策一旦作出，就不可再更改. 在某些情况下，贪婪算法得到的可能是最优解，但更多的情况下，可能只是近似解. 作出贪婪决策的依据称为贪婪准则(greedy criterion). 例如，平时购物找钱时，为使找回的零钱的硬币数最少，通常不考虑找零钱的所有方案，而是从最大面值的币种开始，按递减的顺序考虑各币种，先尽量用大面值的币种，当不足大面值币种的金额时才去考虑下一种较小面值的币种. 这就是在使用贪婪算法. 这种方法在现实中总是最优，因为现实中银行对其发行的硬币种类和硬币面值进行了巧妙的安排. 但如果只有面值分别为 1,5 和 11 单位的硬币，而希望找回总额为 15 单位的硬币，按贪婪算法，应找 1 个 11 单位面值的硬币和 4 个 1 单位面值的硬币，共找回 5 个硬币. 但最优的解应是 3 个 5 单位面值的硬币. 所以算法并不保证在所有的情况下都得到最优结果，然而，贪婪算法却是一种具有直觉的倾向且通常情况下所得结果总是非常接近最优值的. 它利用的规则就是在实际环境中人们所采用的规则. 这种算法也称为启发式方法(heuristics method).

对于一些除了"穷举"方法外没有有效算法的问题，用贪婪算法往往能很快地得出较好的结果，如果此较好结果与最优结果相差不是很多的话，此方法还是很实用的. 对于 NP 完全问题，这种方法还是经常考虑的. 启发式方法在数学建模竞赛中的使用是非常普遍的.

2. 装箱问题(bin packing problem，简称 BPP)

装箱问题的数学模型：当你装一个箱子时，你会发现要使箱子尽可能装满不是一件很容易的事，你往往需要作些调整. 从理论上讲，装箱问题是一个很难的组合优化问题，即使用计算机也是不容易解决的.

装箱问题表述如下：设有编号为 $1,2,\cdots,n$ 的 n 种物品，体积分别为 $\omega_1,\omega_2,\cdots,\omega_n$. 将这 n 种物品装到容量都为 V 的若干箱子里(更一般的装箱问题还可以要求容量不是相同的). 约定这 n 种物品的体积均不超过 V，即对于 $1\leqslant j\leqslant n$，有 $0<\omega_j\leqslant V$. 不同的装箱方案所需要的箱子数目可能不同. 求使装尽这 n 种物品的箱子数最少的

装箱方案.

若找出将 n 种物品的集合划分成 n 个或小于 n 个子集的所有划分,最优解就可以找到. 但所有可能划分的总数太大. 这时,找出所有可能的划分要花费的时间也许是无法承受的.

我们可以设存在一组变量 y_i,其可能取值为0或1. 例如,y_i 为1,则箱子 i 装入物品,反之表示箱子 i 空着;用 x_{ij} 为1表示物品 j 装入箱子 i,反之表示物品 j 未放入箱子 i. 我们的目的就是找到一组 y_i,使它满足装入该箱的物品重量不超过 V. 满足限制条件的每一组 y_i 都是一个可行解,相应的优化函数是 $\sum_{i=1}^{n} y_i$. 能使 $\sum_{i=1}^{n} y_i$ 取得最小值的方案是最优解.

装箱问题的数学模型表示如下:

$$\min z(y) = \sum_{i=1}^{n} y_i$$

$$\text{s. t.} \begin{cases} \sum_{j=1}^{n} \omega_j x_{ij} \leqslant V y_i, & i \in \mathbf{N} = \{1,2,\cdots,n\} \\ \sum_{i=1}^{n} x_{ij} = 1 & \text{(一件物品装在一个箱子里)} \\ y_i = 0 \text{ 或 } 1, & i \in \mathbf{N} \\ x_{ij} = 0 \text{ 或 } 1, & i,j \in \mathbf{N} \end{cases}$$

其中,$y_i = 1$ 表示箱子 i 装入物品,反之表示箱子 i 空着;$x_{ij} = 1$ 表示物品 j 装入箱子 i,反之表示物品 j 未放入箱子 i;ω_j 表示第 j 个物品的体积.

装箱问题是经典的NP难解问题,这意味着该问题不存在在多项式时间内求得精确解的算法(如果 $\mathrm{P} \neq \mathrm{NP}$),因此对装箱问题算法的研究指的是对其近似算法的研究,所谓近似算法即该算法可以求得与精确解接近的结果,但不一定得到精确解. 为此,对装箱问题可采用非常简单的近似算法,即贪婪算法. 该算法依次将物品放到它第一个能放进去的箱子中,该算法虽不能保证找到最优解,但还是能找到非常好的解.

装箱问题中最早被研究的是一维装箱问题. 随着研究的深入,人们发现实际生活中更多存在的是一些带约束的装箱问题,因此也就提出了诸如二维装箱问题(条形装箱问题、剪裁问题)、三维装箱问题、变容装箱问题、有色装箱问题、对偶装箱问题等一系列带约束的装箱问题. 由于这些问题与生俱来的复杂性,虽然已经有一些研究成果发表了,但是其研究还是相当的困难.

3. 案例分析——一维0/1背包问题

例6.3 有 N 件物品和一个容量为 W 的背包. 第 j 件物品的价值是 c_j,体积是 w_j. 求将哪些物品装入背包可在满足背包容量允许的前提下使价值总和最大.

分析　设 x_j 为二进制变量,如果物品 j 被放入背包,则 $x_j = 1$,否则 $x_j = 0$,背包问题的数学模型描述如下:

$$\max \sum_{j=1}^{n} c_j x_j$$

$$\text{s. t. } \sum_{j=1}^{n} w_j x_j \leqslant W$$

$$x_j \in \{0,1\}, \quad j = 1,2,\cdots,n$$

这是最基础的背包问题,特点是:每种物品仅有一件,可以选择放或不放. 问题自身的特性决定了该问题运用贪婪算法可以得到最优解或较优解. 通常这里有三种贪婪准则:

(1) 质量贪婪准则. 从剩下的物品中选择可装入背包的重量最小的物品,重复此过程直至不满足条件为止. 在一般情况下也不一定能得到最优解. 考虑 $n=2$, $\boldsymbol{w}=[10,20]$, $\boldsymbol{c}=[5,100]$, $W=25$. 当利用重量贪婪策略时,获得的解为 $\boldsymbol{x}=[1,0]$,比最优解[0,1]要差.

(2) 价值贪婪准则. 从剩余的物品中选择可装入背包的价值最大的物品,利用这种规则,价值最大的物品首先被装入(假设有足够容量),然后是下一个价值最大的物品,重复此过程直至不满足条件为止. 这种策略不能保证得到最优解. 例如,考虑 $n=2$, $\boldsymbol{w}=[100,10,10]$, $\boldsymbol{c}=[20,15,15]$, $W=105$. 当利用价值贪婪准则时,获得的解为 $\boldsymbol{x}=[1,0,0]$,这种方案的总价值为 20. 而最优解为[0,1,1],其总价值为 30.

(3) 价值密度贪婪准则. 从剩下的物品中选择可装入背包的单位价值 c_j/w_j 最大的物品,即按 c_j/w_j 非递增的次序装入物品,只要正被考虑的物品装得进就装入背包,这种策略可能会得到最优解. 这里采取第三种贪婪策略.

算法描述

(1) 输入物品个数 n,背包的容量 limit$W$,每个物品的重量 w_j 和价值 c_j.

(2) 对物品按单位价值 c_j/w_j 从大到小排序.

(3) 将排序后的物品依次装入背包. 对于当前物品 j,若背包剩余可装重量大于或等于 w_j,则将物品 j 装入背包,继续考虑下一个物品 $j+1$.

重复步骤(3),直到得到问题的解,输出.

程序实现详见华中数学建模网 http://www.shumo.cn/大学数学实验栏目.

结果分析

```
Y=

    1  1  0  1  0  1  0  1  1  1  1  0  1  1  0  1
    1  0  1  1  0  1  1  1  1  1  1  1  0  1  0  0
    0  0  1  0  1  0  0  1  1  0  0  0  0  0  0  0
    0  0
```

```
totalw=996
totalc=3095
```

背包问题是经典的NP-hard组合优化问题之一，在经济管理、资源分配、投资决策、装载设计等领域有着重要的应用价值.

4. 贪婪算法的两大要素

从上面的讨论可知，贪婪算法用于求解一类最优化问题，它使用多步决策求解方法，根据选定的最优度量标准(贪婪准则)，每次确定问题的一个分量. 由于贪婪算法每一步上作的选择只是当时的最佳选择，所以并不一定总能产生最优解. 然而，对于一个最优问题，如何才能得知该问题可用贪婪算法求解? 一般来说，适于使用贪婪算法求解的问题都具有下面两大特征：最优度量标准和最优子结构. 最优量度标准是指可以根据该量度标准，实行多步决策进行求解，虽然在该量度意义下所作的这些选择都是局部最优的，但最终得到的解却是全局最优的. 选择合适的量度标准是贪婪算法的核心问题. 值得注意的是，贪婪算法每一步作出的选择可以依赖于以前作出的选择，但决不依赖将来的选择，也不依赖于子问题的解. 虽然贪婪算法的每次选择也是将问题简化为一个规模更小的子问题，但由于贪婪算法某一步的选择并不依赖子问题的求解，每一步选择只按最优量度标准进行. 所以，对于一个贪婪算法，必须证明所采用的量度标准能够导致一个整体最优解. 最优子结构是关于问题最优解的特性. 当一个问题的最优解中包含子问题的最优解时，则称该问题具有最优子结构特性. 一般而言，如果一个最优化问题的解结构具有元组形式(可以表示成向量)，并具有最优子结构特性，就可以尝试选择最优量度准则用贪婪算法求解. 并非所有具有最优子结构的问题，都能够找到合适的最优量度准则，此时我们可以选择使用动态规划.

6.3.2 模拟退火算法

1. 算法简介

模拟退火算法(simulated annealing)得益于材料的统计力学的研究成果. 统计力学表明材料中粒子的不同结构对应于粒子的不同能量水平. 在高温条件下，粒子的能量较高，可以自由运动和重新排列. 在低温条件下，粒子能量较低. 如果从高温开始，非常缓慢地降温(这个过程被称为退火)，粒子就可以在每个温度下达到热平衡. 当系统完全被冷却时，最终形成处于低能状态的晶体. 如果用粒子的能量定义材料的状态，Metropolis算法用一个简单的数学模型描述了退火过程. 假设材料在状态i之下的能量为$E(i)$，那么材料在温度T时从状态i进入状态j就遵循如下规律：

(1) 如果$E(j)\leqslant E(i)$，接受该状态被转换.

(2) 如果$E(j)>E(i)$，则状态转换以如下概率被接受：

$$e^{\frac{E(i)-E(j)}{KT}}$$

其中 K 是物理学中的玻尔兹曼常量，T 是材料温度. 在某一个特定温度下，进行了充分的转换之后，材料将达到热平衡. 这时材料处于状态 i 的概率满足玻尔兹曼分布：

$$P_T\{x=i\}=\frac{e^{-\frac{E(i)}{KT}}}{\sum\limits_{j\subset S}e^{-\frac{E(j)}{KT}}}$$

其中 x 表示材料当前状态的随机变量，S 表示状态空间集合. 显然

$$\lim_{T\to\infty}\frac{e^{-\frac{E(i)}{KT}}}{\sum\limits_{j\subset S}e^{-\frac{E(j)}{KT}}}=\frac{1}{|S|}$$

其中 $|S|$ 表示集合 S 中状态的数量. 这表明所有状态在高温下具有相同的概率，而当温度下降时，

$$\begin{aligned}\lim_{T\to 0}\frac{e^{-\frac{E(i)-E_{\min}}{KT}}}{\sum\limits_{j\subset S}e^{-\frac{E(j)-E_{\min}}{KT}}}&=\lim_{T\to 0}\frac{e^{-\frac{E(i)-E_{\min}}{KT}}}{\sum\limits_{j\subset S}e^{-\frac{E(j)-E_{\min}}{KT}}+\sum\limits_{j\notin S}e^{-\frac{E(j)-E_{\min}}{KT}}}\\&=\lim_{t\to 0}\frac{e^{-\frac{E(i)-E_{\min}}{KT}}}{\sum\limits_{j\subset S_{\min}}e^{-\frac{E(i)-E_{\min}}{KT}}}=\begin{cases}\frac{1}{|S_{\min}|}, & i\subset S_{\min}\\0, & \text{其他}\end{cases}\end{aligned}\tag{6.3.1}$$

其中 $$E_{\min}=\min_{j\subset S}E(j)\quad 且\quad S_{\min}=\{i\,|\,E(i)=E_{\min}\}$$

式(6.3.1)表明当温度降至很低时，材料会以很大概率进入最小能量状态. 假定我们要解决的问题是一个寻找最小值的优化问题. 将物理学中模拟退火的思想应用于优化问题就可以得到模拟退火寻优方法. 考虑这样一个组合优化问题：优化函数为 $f:x\to\mathbf{R}^+$，其中 $x\in S$，它表示优化问题的一个可行解，$\mathbf{R}^+=\{y\,|\,y\in\mathbf{R},y>0\}$，$S$ 表示函数的定义域. $N(x)\in S$ 表示 x 的一个邻域集合.

首先给定一个初始温度 T_0 和该优化问题的一个初始解 $x(0)$，并由 $x(0)$ 生成下一个解 $x'\in N(x(0))$，是否接受 x' 作为一个新解 $x(1)$ 依赖于下面概率：

$$P\{x(0)\to x'\}=\begin{cases}1, & f(x')<f[x(0)]\\e^{-\frac{f(x')-f[x(0)]}{T_0}}, & \text{其他}\end{cases}$$

换句话说，如果生成的解 x' 的函数值比前一个解的函数值更小，则接受 $x(1)=x'$ 作为一个新解；否则以概率 $e^{-\frac{f(x')-f[x(0)]}{T_0}}$ 接受 x' 作为一个新解. 一般地，对于某一个温度 T_i 和该优化问题的一个解 $x(k)$，可以生成 x'. 接受 x' 作为下一个新解 $x(k+1)$ 的概率为

$$P\{x(k)\to x'\}=\begin{cases}1, & f(x')<f[x(k)]\\e^{-\frac{f(x')-f[x(k)]}{T_0}}, & \text{其他}\end{cases}$$

在温度 T_i 下，经过很多次的转移之后，降低温度 T_i，得到 $T_{i+1}<T_i$. 在 T_{i+1} 下重复上述过程. 因此整个优化过程就是不断寻找新解和缓慢降温的交替过程. 最终的

解是对该问题寻优的结果. 我们注意到,在每个 T_i 下,所得到的一个新状态 $x(k+1)$ 完全依赖于前一个状态 $x(k)$,可以和前面的状态 $x(0),L,x(k-1)$ 无关,因此这是一个马尔可夫过程. 使用马尔可夫过程对上述模拟退火的步骤进行分析,结果表明:从任何一个状态 $x(k)$ 生成新状态 x 的概率,在 $N(x(k))$ 中是均匀分布的,且新状态 x' 被接受的概率满足式(6.3.1),那么经过有限次的转换,在温度 T_i 下的平衡态 x_i 的分布由下式给出:

$$P_i(T_i)=\frac{e^{-\frac{f(x_i)}{T}}}{\sum_{j\subset S}e^{-\frac{f(x_j)}{T_i}}}$$

当温度 T 降为 0 时,x_i 的分布为

$$p_i^*=\begin{cases}\frac{1}{|S_{\min}|}, & x_i\subset S_{\min}\\ 0, & \text{其他}\end{cases} \quad 且 \quad \sum_{x_i\subset S_{\min}} p_i^*=1$$

这说明如果温度下降十分缓慢,而在每个温度都有足够多次的状态转移,使之在每一个温度下达到热平衡,则全局最优解将以概率 1 被找到. 因此可以说模拟退火算法可以找到全局最优解.

在模拟退火算法中应注意以下问题:

(1) 理论上,降温过程要足够缓慢,要使得在每一温度下达到热平衡. 但在计算机实现中,如果降温速度过缓,所得到的解的性能会较为令人满意,但是算法会太慢,相对于简单的搜索算法不具有明显优势. 如果降温速度过快,很可能最终得不到全局最优解. 因此使用时要综合考虑解的性能和算法速度,在两者之间采取一种折中.

(2) 要确定在每一温度下状态转换的结束准则. 实际操作可以考虑当连续 m 次的转换过程没有使状态发生变化时结束该温度下的状态转换. 最终温度的确定可以提前定为一个较小的值 Te,或连续几个温度下转换过程没有使状态发生变化算法就结束.

(3) 选择初始温度和确定某个可行解的邻域的方法也要恰当.

2. 案例分析

例 6.4 已知敌方 100 个目标的经度、纬度见表 6.7.

表 6.7 经度和纬度数据表

经度	纬度	经度	纬度	经度	纬度	经度	纬度
53.7121	15.3046	51.1758	0.0322	46.3253	28.2753	30.3313	6.9348
56.5432	21.4188	10.8198	16.2529	22.7891	23.1045	10.1584	12.4819
20.105	15.4562	1.9451	0.2057	26.4951	22.1221	31.4847	8.964
26.2418	18.176	44.0356	13.5401	28.9836	25.9879	38.4722	20.1731
28.2694	29.0011	32.191	5.8699	36.4863	29.7284	0.9718	28.1477

续表

经度	纬度	经度	纬度	经度	纬度	经度	纬度
8.9586	24.6635	16.5618	23.6143	10.5597	15.1178	50.2111	10.2944
8.1519	9.5325	22.1075	18.5569	0.1215	18.8726	48.2077	16.8889
31.9499	17.6309	0.7732	0.4656	47.4134	23.7783	41.8671	3.5667
43.5474	3.9061	53.3524	26.7256	30.8165	13.4595	27.7133	5.0706
23.9222	7.6306	51.9612	22.8511	12.7938	15.7307	4.9568	8.3669
21.5051	24.0909	15.2548	27.2111	6.207	5.1442	49.243	16.7044
17.1168	20.0354	34.1688	22.7571	9.4402	3.92	11.5812	14.5677
52.1181	0.4088	9.5559	11.4219	24.4509	6.5634	26.7213	28.5667
37.5848	16.8474	35.6619	9.9333	24.4654	3.1644	0.7775	6.9576
14.4703	13.6368	19.866	15.1224	3.1616	4.2428	18.5245	14.3598
58.6849	27.1485	39.5168	16.9371	56.5089	13.709	52.5211	15.7957
38.43	8.4648	51.8181	23.0159	8.9983	23.644	50.1156	23.7816
13.7909	1.951	34.0574	23.396	23.0624	8.4319	19.9857	5.7902
40.8801	14.2978	58.8289	14.5229	18.6635	6.7436	52.8423	27.2880
39.9494	29.5114	47.5099	24.0664	10.1121	27.2662	28.7812	27.6659
8.0831	27.6705	9.1556	14.1304	53.7989	0.2199	33.6490	0.3980
1.3496	16.8359	49.9816	6.0828	19.3635	17.6622	36.9545	23.0265
15.7320	19.5697	11.5118	17.3884	44.0398	16.2635	39.7139	28.4203
6.9909	23.1804	38.3392	19.9950	24.6543	19.6057	36.9980	24.3992
4.1591	3.1853	40.1400	20.3030	23.9876	9.4030	41.1084	27.7149

(1) 我方有一个基地，经度和纬度为(70,40). 假设我方飞机的速度为 1000 km/h. 我方派一架飞机从基地出发，侦察完敌方所有目标，再返回原来的基地. 在敌方每一目标点的侦察时间不计，求该架飞机所花费的时间(假设我方飞机巡航时间可以充分长).

(2) 我方有三个基地，经度、纬度分别为(70,40)，(72,45)，(68,48). 假设我方所有无人侦察机的速度都为 1000 km/h. 三个基地各派出一架飞机侦察敌方目标，怎样划分任务，才能使时间最短，且任务比较均衡?

分析 这是一个旅行商问题. 我们给基地编号为 1，敌方目标依次编号为 2，3，…，101，最后我方基地再重复编号为 102(这样便于程序中计算). 距离矩阵 $\boldsymbol{D}=(d_{ij})_{102\times102}$，其中 d_{ij} 表示表示 i,j 两点的距离，$i,j=1,2,\cdots,102$，这里 $\boldsymbol{D}$ 为实对称矩阵. 则问题是求一个从点 1 出发，走遍所有中间点，到达点 102 的一个最短路径. 上面问题中给定的是地理坐标(经度和纬度)，我们必须求两点间的实际距离. 设 A，B 两点的地理坐标分别为(x_1,y_1)，(x_2,y_2)，过 A，B 两点的大圆的劣弧长即为两点

的实际距离. 以地心为坐标原点 O,以赤道平面为 XOY 平面,以 0 度经线圈所在的平面为 XOZ 平面建立三维直角坐标系. 则 A,B 两点的直角坐标分别为

$$A(R\cos x_1 \cos y_1, R\sin x_1 \cos y_1, R\sin y_1)$$
$$B(R\cos x_2 \cos y_2, R\sin x_2 \cos y_2, R\sin y_2)$$

其中 $R=6370$ 为地球的半径. A,B 两地的实际距离为

$$d=R\arccos\left(\frac{\overrightarrow{OA}\times\overrightarrow{OB}}{|\overrightarrow{OA}|\times|\overrightarrow{OB}|}\right)$$

化简得 $$d=R\arccos[\cos(x_1-x_2)\cos y_1\cos y_2+\sin y_1\sin y_2]$$

求解的模拟退火算法描述如下:

(1) 解空间. 解空间 S 可表为$\{1,2,\cdots,102\}$的所有固定起点和终点的循环排列集合,即

$S=\{(\pi_1,\pi_2,\cdots,\pi_{102})|\pi_1=1,(\pi_2,\pi_3,\cdots,\pi_{101})$为$\{2,3,\cdots,101\}$的循排列,$\pi_{102}=102\}$ 其中每一个循环排列表示侦察 100 个目标的一个回路,$\pi_i=j$ 表示在第 i 次侦察 j 点,初始解可选为$(1,2,\cdots,102)$,本题中我们使用 Monte Carlo 方法求得一个较好的初始解.

(2) 目标函数. 此时的目标函数为侦察所有目标的路径长度(或称代价函数). 我们要求:

$$\min f(\pi_1,\pi_2,\cdots,\pi_{102})=\sum_{i=1}^{101}d_{\pi_i\pi_{i+1}}$$

(3) 新解的产生.

① 2 变换法:任选序号 $u,v(u<v)$ 交换 u 与 v 之间的顺序,此时的新路径为

$$\pi_1\cdots\pi_{u-1}\pi_v\pi_{v-1}\cdots\pi_{u+1}\pi_u\pi_{v+1}$$

② 3 变换法:任选序号 u,v 和 w,将 u 和 v 之间的路径插到 w 之后,对应的新路径为(设 $u<v<w$)

$$\pi_1\cdots\pi_{u-1}\pi_{v+1}\cdots\pi_w\pi_u\cdots\pi_v\pi_{w+1}\cdots\pi_{102}$$

(4) 代价函数差 Δf. 对于 2 变换法,路径差可表示为

$$\Delta f=(d_{\pi_{u-1}\pi_v}+d_{\pi_u\pi_{v+1}})-(d_{\pi_{u-1}\pi_u}+d_{\pi_v\pi_{v+1}})$$

(5) 接受准则.

$$p=\begin{cases}1, & \Delta f<0\\ \exp\{-\Delta f/T\}, & \Delta f\geqslant 0\end{cases}$$

如果 $\Delta f<0$,则接受新的路径. 否则,以概率 $\exp\{-\Delta f/T\}$ 接受新的路径,即若 $\exp\{-\Delta f/T\}$ 大于 0 到 1 之间的随机数则接受.

(6) 降温. 利用选定的降温系数 α 进行降温,即 $T\to\alpha T$,得到新的温度,这里我们取 $\alpha=0.999$.

(7) 结束条件. 用选定的终止温度 $e=10^{-30}$,判断退火过程是否结束. 若 $T<e$,算

法结束，输出当前状态.

程序实现详见华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目.

3. 结果分析

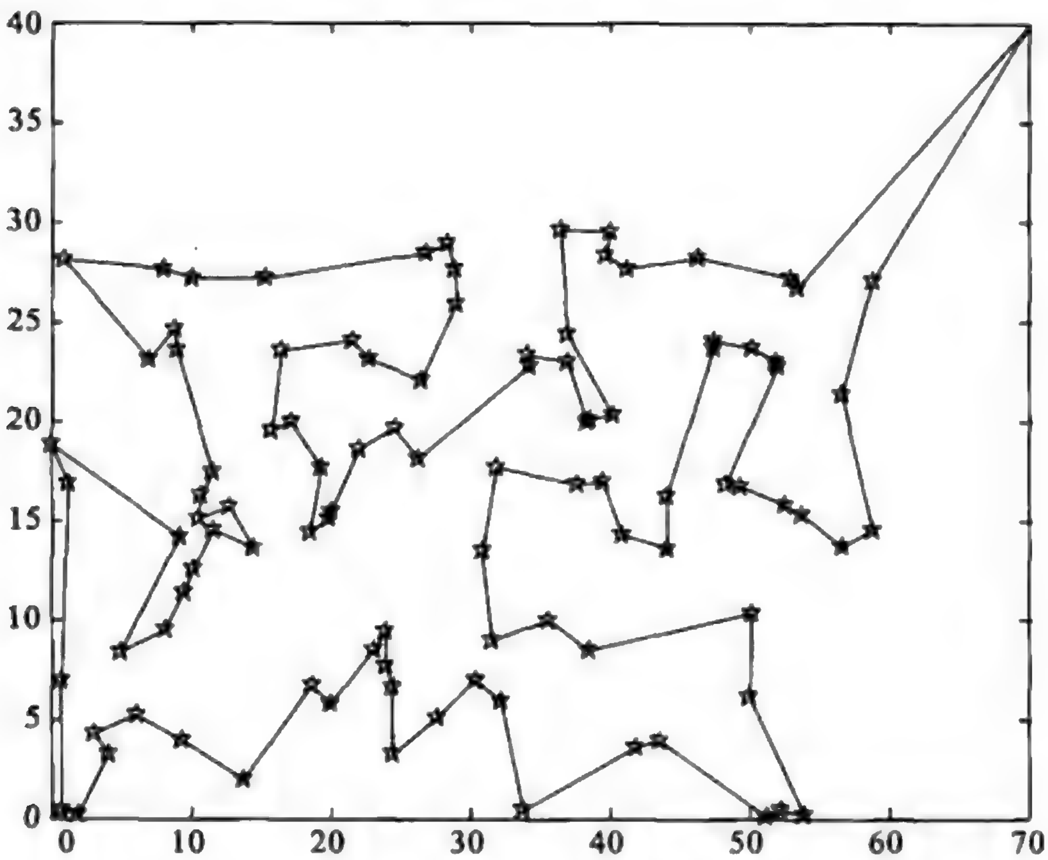


图 6.11　最近路径航线

大致线路如图 6.11 所示，回路之间没有交叉点，这也反映了我们的算法是有效的. 路径点如下：

1　17　3　45　67　2　92　82　48　72　14　27　10　84　18

97　65　94　70　19　63　62　26　29　34　66　90　86　8　39

78　88　28　57　47　23　58　81　25　68　7　22　71　37　32

12　53　75　33　13　24　49　61　16　91　73　4　41　76　69

11　64　85　77　31　79　40　60　9　5　54　55　96　89　6

56　21　99　101　100　98　44　38　50　80　51　15　42　20　30

74　83　87　59　46　52　93　43　36　35　95　102

4. 评论、体会与展望

1）模拟退火算法的模型

模拟退火算法可以分解为解空间、目标函数和初始解三部分.

模拟退火的基本思想：

(1) 初始化：初始温度 T(充分大)，初始解状态 S(算法迭代的起点)，每个 T 值

的迭代次数 L.

(2) 对 $k=1,2,\cdots$ 进行第(3)～(6)步.

(3) 产生新解 S'.

(4) 计算增量 $\Delta t'=C(S')-C(S)$,其中 $C(S)$ 为评价函数.

(5) 若 $\Delta t'<0$ 则接受 S' 作为新的当前解,否则以概率 $\exp\{-\Delta t'/T\}$ 接受 S' 作为新的当前解.

(6) 如果满足终止条件则输出当前解作为最优解,结束程序. 终止条件通常取为连续若干个新解都没有被接受时终止算法.

(7) T 逐渐减少,且 $T\to 0$,然后转第(2)步.

2) 算法步骤

模拟退火算法新解的产生和接受可分为如下 4 个步骤:

第一步 由一个产生函数从当前解产生一个位于解空间的新解;为便于后续的计算和接受,减少算法耗时,通常选择由当前新解经过简单的变换即可产生新解的方法,如对构成新解的全部或部分元素进行置换、互换等. 注意到产生新解的变换方法决定了当前新解的邻域结构,因而对冷却进度表的选取有一定的影响.

第二步 计算与新解所对应的目标函数差. 因为目标函数差仅由变换部分产生,所以目标函数差的计算最好按增量计算. 事实表明,对大多数应用而言,这是计算目标函数差的最快方法.

第三步 判断新解是否被接受,判断的依据是一个接受准则,最常用的接受准则是 Metropolis 准则:若 $\Delta t'<0$ 则接受 S' 作为新的当前解 S,否则以概率 $\exp\{-\Delta t'/T\}$ 接受 S' 作为新的当前解 S.

第四步 当新解被确定接受时,用新解代替当前解,这只需将当前解中对应于产生新解时的变换部分予以实现,同时修正目标函数值即可. 此时,当前解实现了一次迭代. 可在此基础上开始下一轮试验. 而当新解被判定为舍弃时,则在原当前解的基础上继续下一轮试验.

模拟退火算法与初始值无关,算法求得的解与初始解状态 S(算法迭代的起点)无关;模拟退火算法具有渐近收敛性,已在理论上被证明是一种以概率 1 收敛于全局最优解的全局优化算法;模拟退火算法具有并行性.

3) 模拟退火算法的参数控制问题

模拟退火算法的应用很广泛,可以求解 NP 完全问题,但其参数难以控制,其主要问题有以下三点:

(1) 温度 T 的初始值设置问题. 温度 T 的初始值设置是影响模拟退火算法全局搜索性能的重要因素之一. 初始温度高,则搜索到全局最优解的可能性大,但因此要花费大量的计算时间;反之,则可节约计算时间,但全局搜索性能可能受到影响. 实际应用过程中,初始温度一般需要依据实验结果进行若干次调整.

(2) 退火速度问题. 模拟退火算法的全局搜索性能也与退火速度密切相关. 一般来说,同一温度下的"充分"搜索(退火)是相当必要的,但这需要计算时间. 实际应用中,要针对具体问题的性质和特征设置合理的退火平衡条件.

(3) 温度管理问题. 温度管理问题也是模拟退火算法难以处理的问题之一. 实际应用中,由于必须考虑计算复杂度的切实可行性等问题,常采用如下所示的降温方式:

$$T(t+1)=kT(t)$$

其中 k 为正的略小于 1.00 的常数,t 为降温的次数.

6.3.3 回溯法与分枝定界法

1. 算法简介

对于这样一类问题,其解可以表示成一个 n 元组$(x_0, x_1, \cdots, x_{n-1})$,求满足约束条件下的可行解,或进一步求使目标函数取最大(最小)值的最优解问题. 这样的问题更容易想到的是在约束条件下搜寻解空间,通过搜索状态空间树的状态求问题答案的方法可分为两类:深度优先搜索(DFS)和广度优先搜索(BFS,D-检索),如果在运用搜索算法时使用剪枝函数(约束条件),便成为回溯法(backtracking)和分枝定界法. 一般而言,回溯法的求解目标是在状态空间树中找出满足约束条件的所有解,而分枝定界法的求解目标则是,找出满足约束条件的一个解,或者是在满足约束条件的解中找出最优解. 这两个算法适合于组合优化问题,如 TSP、n-皇后、子集与数、图的着色、0-1 背包等,这些问题都能够通过它们得到较好的解决.

2. 案例分析

例 6.5 求解著名的 n-皇后问题,即在一个 $n\times n$ 的棋盘上摆放 n 个皇后,使其中任意两个皇后都不同列、同行和在一条斜线上.

这个问题能够使用这两个算法来求解,首先,它的解具有 n-元组的形式;其次,问题提供显示约束来确定状态解空间;最后,应能设计有效的约束函数,缩小检索空间(下面只给出使用回溯算法解决的分析和解决方案). 为了使用回溯算法求解 n-皇后问题,首先考虑问题解的结构形式. 由于每个皇后不应该在同一行上,不失一般性,假设将第 i 个皇后放在第 i 行上,这样可用 n 元数组$(x_0, x_1, \cdots, x_{n-1})$便是 n-皇后问题的解,其中 x_i 表示第 i 行的皇后的列号.

对于 n-皇后问题,其显示约束的一种观点是:$S_i=\{0,1,\cdots,n-1\}$,相应的隐式约束为:对任意 $0\leqslant i,j<n$,当 $i\neq j$ 时,$x_i\neq x_j$ 且 $|i-j|\neq|x_i-x_j|$,此时解空间的大小为 n^n. 另一种观点显示约束为:$S_i=\{0,1,\cdots,n-1\}$且 $x_i\neq x_j(0\leqslant i,j<n,i\neq j)$. 相应的隐式约束为:对任意 $0\leqslant i,j<n$,当 $i\neq j$ 时,$|i-j|\neq|x_i-x_j|$,此时相应的解空间大小为 $n!$. 最后来看约束函数的设计. 约束函数从隐式约束产生:当 $0\leqslant i,j<k$,$i\neq j$时,要求 $x_i\neq x_j$ 且 $|i-j|\neq|x_i-x_j|$(这里使用第一种观点).

程序实现详见华中数学建模网 http://www.shumo.cn/大学数学实验栏目.

3. 结果分析

这里求的是 $n=8$ 的情形,由于可行解有 92 种,这里仅给出其中的一种如下:

0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0

6.3.4 禁忌搜索算法

1. 算法简介

禁忌搜索算法(tabusearch)是组合优化算法的一种,是局部搜索算法的扩展. 禁忌搜索算法是人工智能在组合优化算法中的一个成功应用. 禁忌搜索算法的特点是采用了禁忌技术. 所谓禁忌就是禁止重复前面的工作. 禁忌搜索算法用一个禁忌表记录下已经到达过的局部最优点,在下一次搜索中,利用禁忌表中的信息不再或有选择地搜索这些点. 禁忌搜索算法实现的技术问题是算法的关键. 禁忌搜索算法涉及候选集合邻域、禁忌对象和禁忌长度、评价函数、特赦规则、记忆频率信息等概念.

(1) 邻域. 在组合优化中,距离的概念通常不再适用,但是在一点附近搜索另一个下降的点仍然是组合优化数值求解的基本思想. 因此,需要重新定义邻域的概念. 一个组合优化问题可用三参数(D,F,f)表示,其中 D 表示决策变量的定义域,F 表示可行解区域,F 中的任何一个元素称为该问题的可行解,f 表示目标函数. 满足 $f(x^*)=\min\{f(x)\mid x\in F\}$的可行解 x^* 称为该问题的最优解.

定义 6.1 对于组合优化问题(D,F,f),D 上的一个映射:$N:S\in D\rightarrow N(S)\in 2^D$ 称为一个邻域映射,其中 2^D 表示 D 的所有子集组成的集合,$N(S)$称为 S 的邻域,$S'\in N(S)$称为 S 的一个邻居.

(2) 候选集合. 候选集合由邻域中的邻居组成. 常规的方法是从邻域中选择若干个目标值或评价值最佳的邻居入选.

(3) 禁忌对象和禁忌长度. 禁忌表中的两个主要指标是禁忌对象和禁忌长度. 禁忌搜索算法中,由于我们要避免一些操作的重复进行,就要将一些元素放到禁忌表中以禁止对这些元素进行操作,这些元素就是我们指的禁忌对象. 禁忌长度是禁忌对象不允许选取的迭代次数. 一般是给禁忌对象 x 一个数(禁忌长度)t,要求对象 x 在 t 步迭代内被禁,在禁忌表中采用 $\mathrm{tabu}(x)=t$ 记忆,每迭代一步,该项指标做运算

$\text{tabu}(x)=t-1$，直到 $\text{tabu}(x)=0$ 时解禁. 于是，我们可将所有元素分成两类，被禁元素和自由元素. 禁忌长度 t 的选取可以有多种方法，例如，$t=$ 常数，或 $t=[n]$，其中 n 为邻域中邻居的个数. 这种规则容易在算法中实现.

(4) 评价函数. 评价函数是候选集合元素选取的一个评价公式，候选集合的元素通过评价函数值来选取. 以目标函数作为评价函数是比较容易理解的. 目标值是一个非常直观的指标，但有时为了方便或易于计算，会采用其他函数来取代目标函数.

(5) 特赦规则. 在禁忌搜索算法的迭代过程中，会出现候选集中的全部对象都被禁忌，或有一对象被禁，但若解禁则其目标值将有非常大的下降的情况. 在这样的情况下，为了达到全局最优，我们会让一些禁忌对象重新可选. 这种方法称为特赦，相应的规则称为特赦规则.

(6) 记忆频率信息. 在计算的过程中，记忆一些信息对解决问题是有利的. 如一个最好的目标值出现的频率很高，这使我们有理由推测：现有参数的算法可能无法再得到更好的解. 根据解决问题的需要，我们可以记忆解集合、被禁对象组、目标值集合等的出现频率. 频率信息有助于进一步加强禁忌搜索的效率. 我们可以根据频率信息动态控制禁忌的长度. 一个最佳的目标值出现的频率很高，有理由终止计算而将此值认为是最优值.

2. 案例分析

这里继续解决例 6.4.

分析与建模

(1) 求解的禁忌搜索算法描述如下：

① 解空间. 解空间 S 可表为 $\{1,2,\cdots,102\}$ 的所有固定起点和终点的循环排列集合，即

$$S=\{(\pi_1,\pi_2,\cdots,\pi_{102})\mid\pi_1=1,(\pi_2,\pi_3,\cdots,\pi_{101})\text{为}\{2,3,\cdots,101\}\text{的循环排列},\pi_{102}=102\}$$

其中每一个循环排列表示侦察 100 个目标的一个回路，$\pi_i=j$ 表示第 i 次侦察 j 点.

② 目标函数. 目标函数为侦察所有目标的路径长度. 我们要求

$$\min f(\pi_1,\pi_2,\cdots,\pi_{102})=\sum_{i=1}^{101}d_{\pi_i\pi_{i+1}}$$

③ 候选集合.

定义 6.2 对于路径 $\pi_1\cdots\pi_{u-1}\pi_u\pi_{u+1}\cdots\pi_{v-1}\pi_v\pi_{v+1}\cdots\pi_{102}$，交换 u 与 v 之间的顺序，得到的新路径为 $\pi_1\cdots\pi_{u-1}\pi_v\pi_{v-1}\cdots\pi_{u+1}\pi_u\pi_{v+1}\cdots\pi_{102}$ 称为原路径二邻域的一个邻居.

定义 6.3 对于路径 $\pi_1\cdots\pi_u\pi_{u+1}\cdots\pi_{v-1}\pi_v\pi_{v+1}\cdots\pi_w\pi_{w+1}\cdots\pi_{102}$，将 u 和 v 之间的路径插到 w 之后，得到的新路径为 $\pi_1...\pi_{u-1}\pi_{v+1}...\pi_w\pi_v...\pi_v\pi_{w+1}...\pi_{102}$，称为原路径三邻域的一个邻居.

如果要考虑当前解的全部二邻域(或三邻域)的邻居，将面临太大的工作量，因此我们用随机选取的方法每次选取 50 个邻居组成的集合作为候选集合，而将省下的时间作更多次搜索，这样做同样可以保证较高的精确度，同时可以大大提高算法的效率.

④ 禁忌长度与禁忌对象. 对于上述定义的二邻域中的邻居总数为 C_{100}^2，禁忌长度取为 $t=70\approx\sqrt{C_{100}^2}$，将禁忌表设计成一个循环队列，初始化禁忌表 $H=\varnothing$. 从候选集合 C 中选出一个向量 $\boldsymbol{x}$，如果 $\boldsymbol{x}\in H$，并且 H 不满，则把向量 $\boldsymbol{x}$ 添加到禁忌表中；如果 H 已满，则最早进入禁忌表的向量出列，向量 $\boldsymbol{x}$ 进入到出列的位置.

⑤ 评价函数. 可以用目标函数作为评价函数，但是这样每选取一个新的路径都得去计算总时间，计算量比较大. 对于上述二邻域中的邻居作为候选集合，每一个新路径中只有两条边发生了变化，因此将目标函数的差值作为评价函数可以极大地提高算法的效率. 评价函数取为

$$\Delta f=(w_{\pi_{u-1}\pi_v}+w_{\pi_u\pi_{v+1}})-(w_{\pi_{u-1}\pi_u}+w_{\pi_v\pi_{v+1}})$$

其中 Δf 表示目标函数差。

禁忌搜索算法的流程如下：

第一步 选定一个初始解 xnow 及给以禁忌表 $H=\varnothing$.

第二步 若满足停止条件，停止计算；否则，在 xnow 的邻域 N(H，xnow)中选出满足禁忌要求的候选集 Can_N(xnow)，在 Can_N(xnow)中选一个评价值最佳的解 xnext，xnow：=xnext，更新禁忌表 H.

重复第二步.

(2) 对于这个问题，我们的基本想法是，先根据敌方基地的分布特点将敌方的基地大体划分在三个区域之内，并使三架侦察机分别对这三个区域的敌军基地进行侦察，求取各自的最短时间. 然后对任务不均衡区域之中的点做适当调整.

解决问题的步骤如下：

① 划分子图. 要达到比较均衡，应使每架飞机的巡航时间基本相同，由于敌方目标分布较均匀，可以将敌方目标的地理位置图分成面积基本相同的三部分.

② 再对每个子图 $G_i(i=1,2,3)$，分别运用禁忌搜索算法求得其最短侦察路径.

③ 均衡任务. 定义均衡率 η，若 η 接近于 1，则上面划分的任务就可以接受；否则用局部搜索算法，调整子图区域，从而调整各分区内点的个数，直至任务达到均衡.

程序详见华中数学建模网 http://www.shumo.cn/大学数学实验栏目.

3. 评论、体会和展望

禁忌搜索算法是一种全局逐步优化算法，是对人类智力的一种模拟. 该算法通过引入一个灵活的存贮结构和相应的禁忌准则来避免和进行搜索，并通过特赦规则来赦免一些被禁忌的优良状态，进而保证多样化的有效搜索以最终实现全局最优化. 相对于模拟退火和遗传算法，它是一种特点不同的 meta-heuristic 算法. 我们可以明

显地看到，邻域函数、禁忌对象、禁忌表和特赦规则，构成了禁忌搜索算法的关键. 其中，邻域函数沿用局部邻域搜索的思想，用于实现邻域搜索；禁忌表和禁忌对象的设置，体现了算法避免迂回搜索的特点；特赦规则，则是对优良状态的奖励，它是对禁忌策略的一种放松. 需要指出的是，上述算法仅是一种简单的禁忌搜索框架，对各关键环节复杂和多样化的设计则可构造出各种禁忌搜索算法. 同时，算法流程中的禁忌对象，可以是搜索状态，也可以是特定搜索操作，甚至是搜索目标值等. ①在搜索过程中可以接受劣解，因此具有较强的“爬山”能力；②新解不是在当前解的邻域中随机产生，而或是优于“best so far”的解，或是非禁忌的最佳解，因此选取优良解的概率远远大于其他解. 由于 TS 算法具有灵活的记忆功能和特赦规则，并且在搜索过程中可以接受劣解，所以具有较强的“爬山”能力，搜索时能够跳出局部最优解，转向解空间的其他区域，从而增强获得更好的全局最优解的概率，所以 TS 算法是一种局部搜索能力很强的全局迭代寻优算法.

6.3.5 粒子群算法

1. 算法简介

粒子群算法(particle swarm optimization，简称 PSO)是通过模拟鸟群的捕食行为得到的. 设想这样一个场景：一群鸟在随机搜索食物，在这个区域里只有一块食物，所有的鸟都不知道食物在那里，但是他们知道当前的位置离食物还有多远. 那么找到食物的最优策略是什么呢? 最简单有效的就是搜寻目前离食物最近的鸟的周围区域.

PSO 从这种模型中得到启示并用于解决优化问题. PSO 中，每个优化问题的解都是搜索空间中的一只鸟，我们称之为“粒子”. 所有的粒子都有一个由被优化的函数决定的适应值(fitness value)，每个粒子还有一个速度决定它们飞翔的方向和距离. 然后粒子们就追随当前的最优粒子在解空间中搜索.

PSO 初始化为一群随机粒子(随机解)，然后通过迭代找到最优解. 在每一次迭代中，粒子通过跟踪两个“极值”来更新自己. 第一个就是粒子本身所找到的最优解，这个解称为个体极值(pBest). 另一个极值是整个种群目前找到的最优解. 这个极值是全局极值(gBest). 另外，也可以不用整个种群而只是用其中一部分作为粒子的邻居，那么在所有邻居中的极值就是局部极值.

在找到这两个最优值时，粒子根据如下的公式来更新自己的速度和新的位置：

```
v[]=v[]+c1*rand()*(pbest[]-present[])+c2*rand()*(gbest[]-present[]) (a)
present[]=persent[]+v[] (b)
```

其中，v[]是粒子的速度，present[]是当前粒子的位置，pbest[]和 gbest[]如前定义，rand()是介于(0,1)之间的随机数，c1，c2 是学习因子，通常 c1＝c2＝2.

程序的伪代码如下：

```
FOR 每一个粒子
{初始化;}
Do{
      FOR 对每一个粒子{
            计算适应度值;
          If 是硬度值大于历史最佳适应度值
              重置为当前最佳的适应度值;
          }
      标记适应度值最大的粒子,并当做最佳的粒子;
      FOR 每一粒子{
          根据公式(a)计算进化速度;
          根据公式(b)计算进化后的位置;
      }
}While 最优值没有满足或者未进入误差范围;
```

在每一维粒子的速度都会被限制在一个最大速度 V_{max},如果某一维更新后的速度超过用户设定的 V_{max},那么这一维的速度就被限定为 V_{max}.

PSO 的参数设置 从上面的例子可以看到,应用 PSO 解决优化问题的过程中有两个重要的步骤:问题解的编码和适应度函数. PSO 的一个优势就是采用实数编码,不需要像遗传算法一样是二进制编码(或者采用针对实数的遗传操作. 例如,对于问题 $f(x)=x_1^2+x_3^2+x_3^2$ 求解,粒子可以直接编码为(x_1,x_2,x_3),而适应度函数就是 $f(x)$. 接着我们就可以利用前面的过程去寻优. 这个寻优过程是一个迭代过程,中止条件一般为设置为达到最大循环数或者最小错误. PSO 中并没有许多需要调节的参数,下面列出了这些参数以及经验设置.

粒子数:一般取 20~40. 其实对于大部分的问题 10 个粒子已经足够可以取得好的结果,不过对于比较难的问题或者特定类别的问题,粒子数可以取到 100 或 200.

粒子的长度:这是由优化问题决定,就是问题解的长度.

粒子的范围:由优化问题决定,每一维可是设定不同的范围.

V_{max}:最大速度,决定粒子在一个循环中最大的移动距离,通常设定为粒子的范围宽度. 例如,上面的例子里,粒子(x_1,x_2,x_3)中 x_1 属于$[-10,10]$,那么 V_{max}的大小就是 20.

学习因子:c_1 和 c_2 通常等于 2,不过在文献中也有其他的取值,但是一般 c_1 等于 c_2 并且范围在 0 和 4 之间.

中止条件:最大循环数以及最小错误要求. 根据具体问题设定.

粒子群算法包括全局版和局部版两种版本. 前者速度快但有时会陷入局部最优;后者收敛速度慢一点但很难陷入局部最优. 在实际应用中,可以先用全局 PSO 找到

大致的结果，再有局部 PSO 进行搜索.

另外的一个参数是惯性权重，由 Shi 和 Eberhart 提出，有兴趣的可以参考他们 1998 年的论文(题目：A modified particle swarm optimizer).

2. 案例分析

例 6.6 2008 年高教社杯全国数学建模竞赛 A 题，这里只解决圆心的确定. (www. mcm. edu. cn)

分析与建模 由此引出我们解决此题的基本想法：

将靶标上圆的圆心的像点视为食物(粒子群算法中的最优解)，在圆外随机找到一些点视为鸟(粒子群算法中的粒子)，让这些点同鸟寻找食物一样根据当前信息及历史信息逐渐接近最优解，即为靶标上圆的圆心的像.

第一步 要求计算图像上某圆像的几何中心，等价为找到一点，使此点到该圆像各点的距离之和为最小. 目标函数：$\min\sum\limits_{j=0}^{N} s_{ij}$.

由此我们确定的适应度函数为 $f(x,y)=\sum\limits_{j=0}^{N_i}(s_{i0},s_{i1},\cdots,s_{ij})$.

注 其中(x,y)为某一粒子坐标. 当 $f(x,y)$ 的值越大时，其粒子的适应度就越小(方便在程序中使用).

第二步 随机产生 30 个粒子，并将 30 个粒子的数据初始化. 初始化第 i 个粒子的初始位置(x_i,y_i) $(i\in[1,30])$，其中 $0\leqslant x_i\leqslant 1024, 0\leqslant y_i\leqslant 768$，这些数据都由随机产生.

设定常量：

(1) 设定加权系数 w_0,w_1,w_2，分别为 0.3,0.5,0.4.

(2) 设定最高速度 v_m 为一合理的整数，3 ～ 100 之间.

(3) 设定时间间隔 dt，$dt\in[1,10]$.

(4) 初始化第 i 个粒子的速度(vx_i,vy_i)，$vx_i\in[-v_m,v_m]$，$vy_i\in[-v_m,v_m]$.

第三步 迭代计算. 第 k 次迭代计算：

(1) 更新例子的当前速度，计算公式$(k\in[0,20])$：

$$vx_i^{(k+1)}=w_0vx_i^{(k)}+w_1(gx_i^{(k)}-vx_i^{(k)})+w_2(ox_i^{(k)}-vx_i^{k})$$

$$vy_i^{(k+1)}=w_0vy_i^{(k)}+w_1(gy_i^{(k)}-vy_i^{(k)})+w_2(oy_i^{(k)}-vy_i^{k})$$

(2) 更新粒子的当前位置，计算公式：

$$x_i^{(k+1)}=x_i^k+vx_i^{k+1}\mathrm{d}t,\qquad y_i^{(k+1)}=y_i^k+vy_i^{k+1}\mathrm{d}t;$$

(3) 更新粒子历史最优位置. 使用度函数判断当前位置与历史最优位置的适应度大小关系. 如果当前位置使用度更高，将历史最优位置更新为当前位置的坐标.

(4) 粒子最优位置更新为群中适应度最高的粒子位置.

第四步 当迭代 20 次之后，得到该圆的几何中心坐标即为$(g_x^{(20)},g_y^{(20)})$. 寻找几

何中心算法结束.

编程实现具体代码详见华中数学建模网 http://www.shumo.cn/大学数学实验栏目.

3. 结果分析

在 MATLAB 中运行程序得圆心的坐标如下:

323	187
422	196
639	212
287	500
583	502

这样我们就得到了圆心的几何坐标,但是我们寻求的全局最优解,是否对所有点都有较好的结果还需要进一步分析,可以另外设置较好的评价函数进行进一步优化,这里不予讨论.

4. 评论、体会和展望

粒子群算法是一种群体协作算法,即每个粒子有自己的行为(移动方向、移动速度),并对自己的历史最优位置、当前位置以及群体中最优位置有所感知,或者称为记忆.由这些行为特征及记忆和感知所得到的信息,通过加权计算以确定下一时刻的移动方向和速度.

粒子群由于有着相互影响及引导的协同合作计算能力,故能使整个群体中的最优粒子快速逼近坐标中的最优位置.粒子群算法有很多的变种,结合牛顿力学或者量子力学等内容,已经发展成为一个较为优秀的全局搜索算法,有兴趣的读者可以查阅更多的资料.

第7章 微分方程与差分方程模型

7.1 微分方程模型

7.1.1 模型的使用背景

在研究实际问题时，常常会联系到某些变量的变化率或导数，这样所得到变量之间的关系式就是微分方程模型. 微分方程模型反映的是变量之间的间接关系，因此，要得到直接关系，就需要求解微分方程.

求解微分方程有三种方法：①求精确解；②求数值解（近似解）；③定性理论方法.

7.1.2 微分方程模型的建立方法

(1) 根据规律列方程. 利用数学、力学、物理、化学等学科中的定理或经过实验检验的规律等来建立微分方程模型.

(2) 微元分析法. 利用已知的定理与规律寻找微元之间的关系式，与第一种方法不同的是对微元而不是直接对函数及其导数应用规律.

(3) 模拟近似法. 在生物、经济等学科的实际问题中，许多现象的规律性不很清楚，即使有所了解也是极其复杂的，建模时在不同的假设下去模拟实际的现象，建立能近似反映问题的微分方程，然后从数学上求解或分析所建方程及其解的性质，再去同实际情况对比，检验此模型能否刻画、模拟某些实际现象.

微分方程建模是数学建模的重要方法，在科技工程、经济管理、生态环境、人口、交通等领域中有着广泛的应用. 本章通过几个案例详细地介绍微分方程建模的方法.

7.1.3 案例分析

1. 缉私问题

一艘缉私舰雷达发现距 c km 处有一艘走私船正以匀速 a km/min 沿直线行驶. 缉私舰立即以最大的速度 b km/min 追赶，若用雷达进行跟踪，保持船的瞬时速度方向始终指向走私船，试求缉私舰追逐路线和追上的时间.

模型建立 建立如图 7.1 所示的坐标系，缉私船在 $(c,0)$ 处发现走私船在 $(0,0)$ 处，走私船逃跑方向为 y 轴方向. 在 t 时刻，走私船到达 $R(0,at)$，缉私舰到达 $D(x,y)$，根据题意有如下关系式：

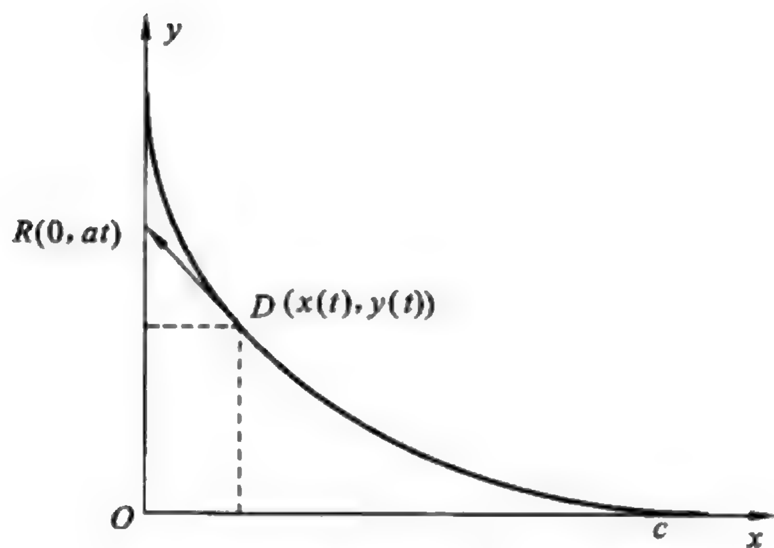


图 7.1　缉私艇追击示意图

$$\frac{\mathrm{d}y}{\mathrm{d}x}=\tan\alpha=\frac{y-at}{x-0}$$

化简，得

$$x\frac{\mathrm{d}y}{\mathrm{d}x}-y=at$$

对 t 求导，得

$$\frac{\mathrm{d}x}{\mathrm{d}t}\cdot\frac{\mathrm{d}y}{\mathrm{d}x}+x\frac{\mathrm{d}^2y}{\mathrm{d}t^2}\cdot\frac{\mathrm{d}x}{\mathrm{d}t}-\frac{\mathrm{d}y}{\mathrm{d}t}=a$$

化简得

$$x\frac{\mathrm{d}^2y}{\mathrm{d}x^2}=-a\frac{\mathrm{d}t}{\mathrm{d}x} \tag{7.1.1}$$

又因$\frac{\mathrm{d}s}{\mathrm{d}t}=b$，$s$ 为弧长，故

$$\frac{\mathrm{d}t}{\mathrm{d}x}=\frac{\mathrm{d}t}{\mathrm{d}s}\cdot\frac{\mathrm{d}s}{\mathrm{d}x}=-\frac{1}{b}\sqrt{1+\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)^2} \tag{7.1.2}$$

由(7.1.1)、(7.1.2)式得

$$\begin{cases}x\dfrac{\mathrm{d}^2y}{\mathrm{d}x^2}=r\sqrt{1+\left(\dfrac{\mathrm{d}y}{\mathrm{d}x}\right)^2}\\ y(c)=0, y'(c)=0\end{cases}$$

其中 $r=a/b$.

模型求解

(1) 求解析解. 令$\frac{\mathrm{d}y}{\mathrm{d}x}=p$，则$\frac{\mathrm{d}^2y}{\mathrm{d}x^2}=\frac{\mathrm{d}p}{\mathrm{d}x}$，

$$\begin{cases}\dfrac{\mathrm{d}p}{\sqrt{1+p^2}}=r\dfrac{\mathrm{d}x}{x}\\ p(c)=0\end{cases}$$

积分得　　$p+\sqrt{1+p^2}=\left(\frac{x}{c}\right)^r$，　　$p-\sqrt{1+p^2}=-\left(\frac{c}{x}\right)^r$

所以
$$\begin{cases}\dfrac{dy}{dx}=\dfrac{1}{2}\left[\left(\dfrac{x}{c}\right)^r-\left(\dfrac{c}{x}\right)^r\right]\\ y(c)=0\end{cases}$$

讨论 ①当 $r=\frac{a}{b}<1$ 时，

$$y=\frac{c}{2}\left[\frac{1}{1+r}\left(\frac{x}{c}\right)^{1+r}-\frac{1}{1-r}\left(\frac{x}{c}\right)^{1-r}\right]+\frac{cr}{1-r^2}$$

当 $x=0$ 时，$y=\frac{cr}{1-r^2}$，

$$t=\frac{y}{a}=\frac{cr}{a(1-r^2)}=\frac{bc}{(b^2-a^2)}$$

$c=3$ km，$a=0.4$ km/min，分别取 $b=0.6$，0.8，1.2 km/min，缉私艇追赶路线图形如图 7.2 所示. 追赶时间分别为 $t=9,5,2.8125$ min.

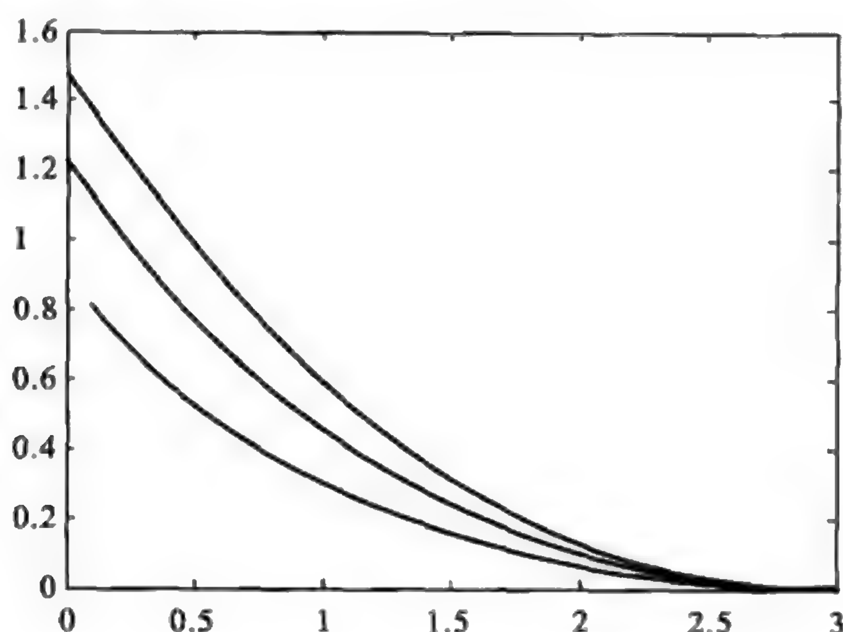


图 7.2 缉私艇追赶路线图

② 当 $r=\frac{a}{b}>1$ 时，

$$y=\frac{c}{2}\left[\frac{1}{1+r}\left(\frac{x}{c}\right)^{1+r}+\frac{1}{r-1}\left(\frac{c}{x}\right)^{r-1}\right]-\frac{cr}{r^2-1}$$

当 $x\to+\infty$ 时，$y\to+\infty$，缉私艇不可能追赶上走私船.

③ 当 $r=1$ 时，

$$y=\frac{1}{2}\left(\frac{x^2-c^2}{2c}-c\ln\frac{x}{c}\right)$$

当 $x\to+\infty$ 时，$y\to+\infty$，缉私艇不可能追赶上走私船.

(2) 求数值解.

假设敌艇逃跑速度 $a=60$ km/h，缉私舰追击速度 $b=80$ km/h，缉私舰发现敌艇时相距 $c=500$ km. 计算得 $k=a/b=0.75$.

首先,建立 M-文件.

```
zhuiji.m
function f=zhuiji(x,y)  %建立微分方程组函数,函数名为'zhuiji'
f=[y(2);0.75*sqrt(1-y(2)^2)/x];
```

其次,建立主程序.

```
zhui.m
[x,y]=ode23('zhuiji',[500,1],[0,0]);  %调用 ode23 求解器求解方程组
plot(x,y(:,1))  %画出图形
```

运行结果如图 7.3 所示.

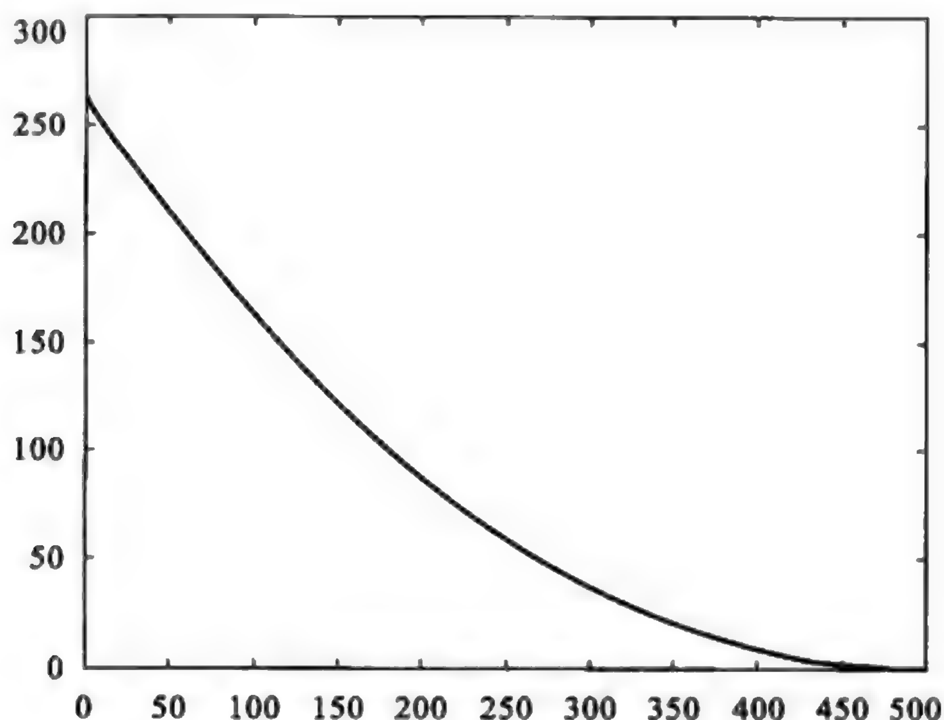


图 7.3　缉私艇追赶路线图

图 7.3 描述了当敌艇逃跑速度 $a=60$ km/h,缉私舰追击速度 $b=80$ km/h,缉私舰发现敌艇时的相距 $c=500$ km 的情况下的追击路线图.

2. 人口增长模型

据考古学家论证,地球上出现生命距今已有 20 亿年,而人类的出现距今却不足 200 万年. 纵观人类人口总数的增长情况,我们发现:1000 年前人口总数为 2.75 亿. 经过漫长的过程,到 1830 年,人口总数达 10 亿;又经过 100 年,在 1930 年,人口总数达 20 亿;30 年之后,在 1960 年,人口总数为 30 亿;又经过 15 年,1975 年的人口总数是 40 亿;12 年之后即 1987 年,人口已达 50 亿.

我们自然会产生这样一个问题:人类人口增长的规律是什么? 如何用数学来描述这一规律?

1) Malthus 模型

1789 年,Malthus 在分析了一百多年的人口统计资料之后,提出了 Malthus 模型.

模型假设

(1) 设 $x(t)$表示 t 时刻的人口数，且 $x(t)$连续可微.

(2) 人口的增长率 r 是常数(增长率＝出生率－死亡率).

(3) 人口数量的变化是封闭的，即人口数量的增加与减少只取决于人口中个体的生育和死亡，且每一个体都具有同样的生育能力与死亡率.

建模与求解 由假设，t 时刻到 $t+\Delta t$ 时刻人口的增量为

$$x(t+\Delta t)-x(t)=rx(t)\Delta t$$

于是得

$$\begin{cases}\dfrac{\mathrm{d}x}{\mathrm{d}t}=rx\\ x(0)=x_0\end{cases}\tag{7.1.3}$$

其解为

$$x(t)=x_0\mathrm{e}^{rt}\tag{7.1.4}$$

模型评价 考虑两百多年来人口增长的实际情况，1961 年世界人口总数为 3.06×10^9，在 1961～1970 年这段时间内，每年平均的人口自然增长率为 2%，则(7.1.4)式可写为

$$x(t)=3.06\times10^9\mathrm{e}^{0.02(t-1961)}\tag{7.1.5}$$

根据 1700～1961 年间世界人口统计数据，我们发现这些数据与(7.1.5)式的计算结果相当符合. 因为在这期间地球上人口大约每 35 年增加 1 倍，而(7.1.5)式算出每 34.6 年增加 1 倍.

利用(7.1.5)式对世界人口进行预测，也会得出惊异的结论：当 $t=2670$ 年时，$x(t)=4.4\times10^{15}$，即 4400 万亿，这相当于地球上每平方米要容纳至少 20 人.

显然，用这一模型进行预测的结果远高于实际人口增长，误差的原因是对增长率 r 的估计过高. 由此，可以对 r 是常数的假设提出疑问.

2) 阻滞增长模型(logistic 模型)

如何对增长率 r 进行修正呢？我们知道，地球上的资源是有限的，它只能提供一定数量的生命生存所需的条件. 随着人口数量的增加，自然资源、环境条件等对人口再增长的限制作用将越来越显著. 如果在人口较少时，我们可以把增长率 r 看成常数，那么当人口增加到一定数量之后，就应当视 r 为一个随着人口的增加而减小的量，即将增长率 r 表示为人口 $x(t)$的函数 $r(x)$，且 $r(x)$为 x 的减函数.

模型假设

(1) 设 $r(x)$为 x 的线性函数，$r(x)=r-sx$(工程师原则，首先用线性).

(2) 自然资源与环境条件所能容纳的最大人口数为 x_m，即当 $x=x_m$ 时，增长率 $r(x_m)=0$.

建模与求解 由假设(1)、(2)可得 $r(x)=r\left(1-\dfrac{x}{x_m}\right)x$，则有

$$\begin{cases}\dfrac{\mathrm{d}x}{\mathrm{d}t}=r\left(1-\dfrac{x}{x_m}\right)x\\ x(t_0)=x_0\end{cases}\tag{7.1.6}$$

(7.1.6)式是一个可分离变量的方程，其解为

$$x(t)=\frac{x_m}{1+\left(\frac{x_m}{x_0}-1\right)e^{-r(t-t_0)}} \tag{7.1.7}$$

模型检验 由(7.1.6)式，计算可得

$$\frac{d^2x}{dt^2}=r^2\left(1-\frac{x}{x_m}\right)\left(1-\frac{2x}{x_m}\right)x \tag{7.1.8}$$

人口总数 $x(t)$ 有如下规律：

(1) $\lim\limits_{x\to+\infty} x(t)=x_m$，即无论人口初值 x_m 如何，人口总数以 x_m 为极限.

(2) 当 $0<x_0<x_m$ 时，$r(x)=r\left(1-\frac{x}{x_m}\right)x>0$，这说明 $x(t)$ 是单调增加的. 又由(7.1.8)式知，当 $x<\frac{x_m}{2}$ 时，$\frac{d^2x}{dt^2}>0$，$x=x(t)$ 为凹函数；当 $x>\frac{x_m}{2}$ 时，$\frac{d^2x}{dt^2}>0$，$x=x(t)$ 为凸函数.

(3) 人口变化率 $\frac{dx}{dt}$ 在 $x=\frac{x_m}{2}$ 时取到最大值，即人口总数达到极限值一半以前是加速生长时期，经过这一点之后，生长速率会逐渐变小，最终达到零.

表 7.1 给出本世纪初用 Malthus 模型和 logistic 模型计算所得的美国人口预测数. 自 1790 年开始，每十年统计一次，取 $x_m=197273000$，$r=0.03134$，人口数量以百万为单位.

表 7.1 美国人口预测表

年份	实际统计资料/百万	Malthus 模型/百万	误差/%	logistic 模型/百万	误差/%
1790	3.929	3.9290	0.0000	3.9290	0.0000
1800	5.308	5.3751	1.2649	5.3360	0.5280
1810	7.240	7.3536	1.5685	7.2281	−0.1645
1820	7.638	10.0602	4.3804	7.7569	1.2340
1830	12.866	13.7630	6.9720	13.1095	1.8927
1840	17.069	18.8287	10.3096	17.5065	2.5632
1850	23.192	25.7590	11.0686	23.1926	0.0024
1860	31.443	35.2401	12.0762	30.4130	−3.2759
1870	38.558	48.2109	25.0347	37.3729	2.1133
1880	50.156	65.9558	31.5013	50.1786	0.0450
1890	62.948	90.2321	43.3438	62.7710	−0.2812
1900	75.995	123.4437	62.4366	76.8720	1.1540
1910	91.972	168.8794	83.6205	91.9746	0.0028

续表

年份	实际统计资料/百万	Malthus 模型/百万	误差/%	logistic 模型/百万	误差/%
1920	105.711	231.0386	118.5569	107.3976	1.5955
1930	122.775	316.0767	157.4439	122.4006	−0.3049
1940	131.669	432.4146	228.4104	136.3205	3.5327
1950	150.697	591.5729	292.5579	148.6799	−1.3385

用(7.1.4)式对 1790 年以来的美国人口进行检验,发现有很大差异,见表 7.1.

利用 MATLAB 求解 Malthus 模型和 logistic 模型,预测美国人口数量,程序如下所示:

```
k=197.273;  %xm=197.273
r=0.03134;  %r=0.03134
t=0:10:160;  %时间间隔为 10 年
n0=3.929;
n1=[3.929 5.308 7.240 7.638 12.866 17.069 23.192 31.4433 8.558 50.156 62.948
    75.9 95 91.972 105.711 122.775 131.669 150.697];  %实际统计资料
n2=n0*exp(r*t);  %Malthus 模型
n3=k./(1+((k/n0)-1).*exp(-r.*t));  %logistic 模型
t=t+1790;
plot(t,n1,'k*-',t,n2,'go-',t,n3)
```

运行结果如图 7.4 所示,图中星号的是 logistic 模型求解的预测值,圆圈的是 Malthus 模型预测值,曲线为实际统计值. 由图 7.4 可以看出,Malthus 模型只是在短时间内是精确的,而 logistic 模型与实际统计值十分吻合. 这说明了 logistic 模型能很好地模拟现实的过程.

3. 传染病模型

随着卫生设施的改善,医疗水平的提高及人类文明的不断发展,诸如霍乱、天花等曾经肆虐全球的传染性疾病已经得到了有效的控制. 但是一些新的、不断变异着的传染病毒却悄悄地向人类袭来. 20 世纪 80 年代,十分险恶的艾滋病毒开始肆虐全球,至今仍在蔓延;2003 年春,来历不明的 SARS 病毒突袭人间,给人们的生命财产带来了极大的危害. 长期以来,建立传染病的数学模型来描述传染病的传播过程、分析受感染人数的变化规律、探索制止传染病蔓延的手段等,一直是有关专家关注的一个热点问题.

不同类型传染病的传播过程有其各自不同的特点,弄清这些特点需要相当多的病理知识. 我们在这里不可能从医学角度一一分析各种传染病的传播特点,而只能按照一般的传播机理来建立数学模型.

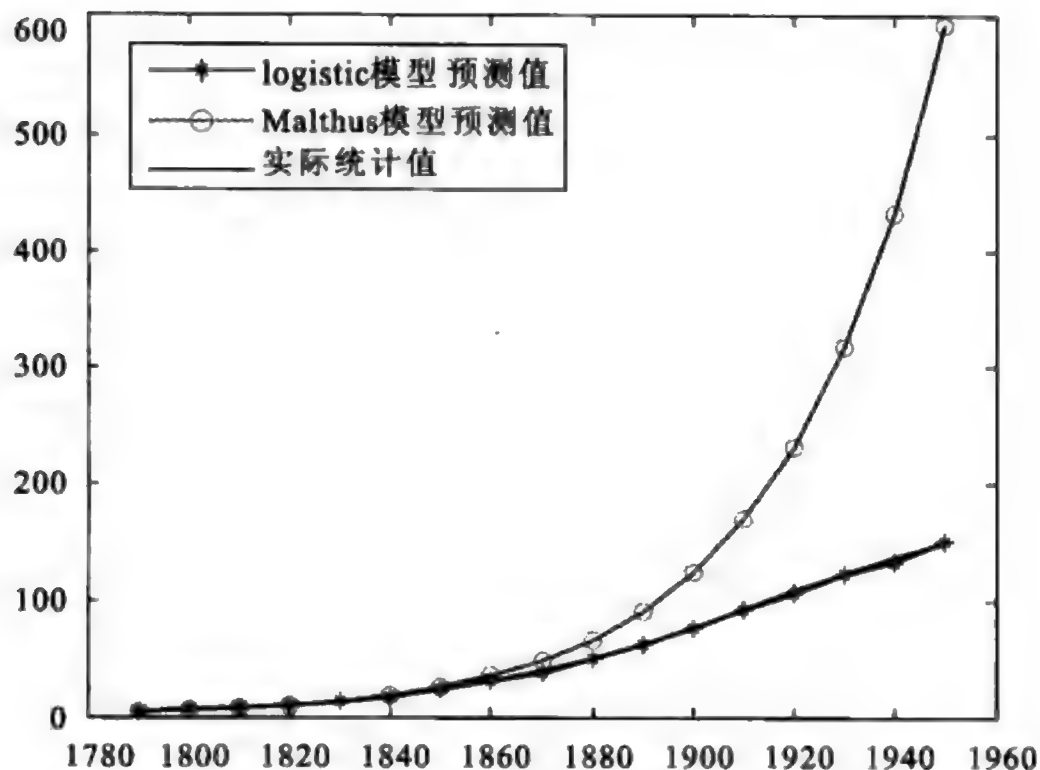


图 7.4　美国人口预测图

1) 模型 I

这是一个最简单的传染病模型. 设时刻 t 的病人人数 $x(t)$ 是连续、可微函数，并且每个病人每天有效接触(足以使人致病的接触)的平均人数是常数 λ. 考察 t 到 $t+\Delta t$ 这段时间内病人人数的增加，于是就有

$$x(t+\Delta t)-x(t)=\lambda x(t)\Delta t$$

$$\frac{x(t+\Delta t)-x(t)}{\Delta t}=\lambda x(t) \tag{7.1.9}$$

再设 $t=0$ 时，有 x_0 个病人. 并对(7.1.9)式取 $\Delta t\to 0$ 时的极限，得微分方程：

$$\frac{\mathrm{d}x}{\mathrm{d}t}=\lambda x, \qquad x(0)=x_0 \tag{7.1.10}$$

方程(7.1.10)的解为

$$x(t)=x_0\mathrm{e}^{\lambda t} \tag{7.1.11}$$

(7.1.11)式表明，随着 t 的增加，病人人数 $x(t)$ 将会无限增长，这显然是不符合实际的.

上述建模失败的原因是：

(1) 在病人有效接触的人群中，有健康人也有病人，而其中只有健康人才可以被传染为病人. 所以在下面改进的模型中必须区别这两种人.

(2) 人群的总人数是有限的，不是无限的. 并且随着病人人数的增加，健康人的人数在逐渐减少. 因此，病人的人数不会无限地增加下去.

2) 模型 II

模型假设

(1) 在疾病传播期内所考察地区的总人数 N 不变，既不考虑生死，也不考虑迁移.

(2) 人群分为易感染者和已感染者两类，以下简称健康者和病人. 并记时刻 t 这两类人在总人数中所占的比例分别为是 $s(t)$ 和 $i(t)$.

(3) 每个病人每天有效接触的平均人数是常数 λ，λ 称为日接触率. 当病人与健康者有效接触时，使健康者受感染变为病人.

根据上述假设，每个病人每天可使 $\lambda s(t)$ 个健康者变为病人. 因为病人人数为 $Ni(t)$，所以每天共有 $\lambda Ns(t)i(t)$ 个健康者被感染. 于是 $\lambda Ns(t)i(t)$ 就是病人人数 $Ni(t)$ 的增加率，即有

$$N\frac{\mathrm{d}i}{\mathrm{d}t}=\lambda Ns(t)i(t) \tag{7.1.12}$$

又因为 $s(t)+i(t)=1$，再记初始时刻($t=0$)病人的比例为 i_0，则

$$\frac{\mathrm{d}i}{\mathrm{d}t}=\lambda i(t)[1-i(t)], \qquad i(0)=i_0 \tag{7.1.13}$$

方程(7.1.13)是 logistic 模型，它的解为

$$i(t)=\frac{1}{1+\left(\frac{1}{i_0}-1\right)\mathrm{e}^{-\lambda t}} \tag{7.1.14}$$

当 $i_0=0.09$，$\lambda=0.1$ 时，用 MATLAB 软件求解：

```
y=dsolve('Dy=0.1*y*(1-y)','y(0)=0.09','x')  %求解此微分方程
ezplot(y,[0,60])  %画出微分方程的图像
ezplot('0.1*y*(1-y)',[0,1])  %画出 y 的导数的图像
```

由(7.1.13)、(7.1.14)式及图 7.5 知，当 $i=\frac{1}{2}$ 时，$\frac{\mathrm{d}i}{\mathrm{d}t}$ 达到最大值 $\left(\frac{\mathrm{d}i}{\mathrm{d}t}\right)_m$，这个时刻为

$$t_m=\lambda^{-1}\ln\left(\frac{1}{i_0}-1\right)$$

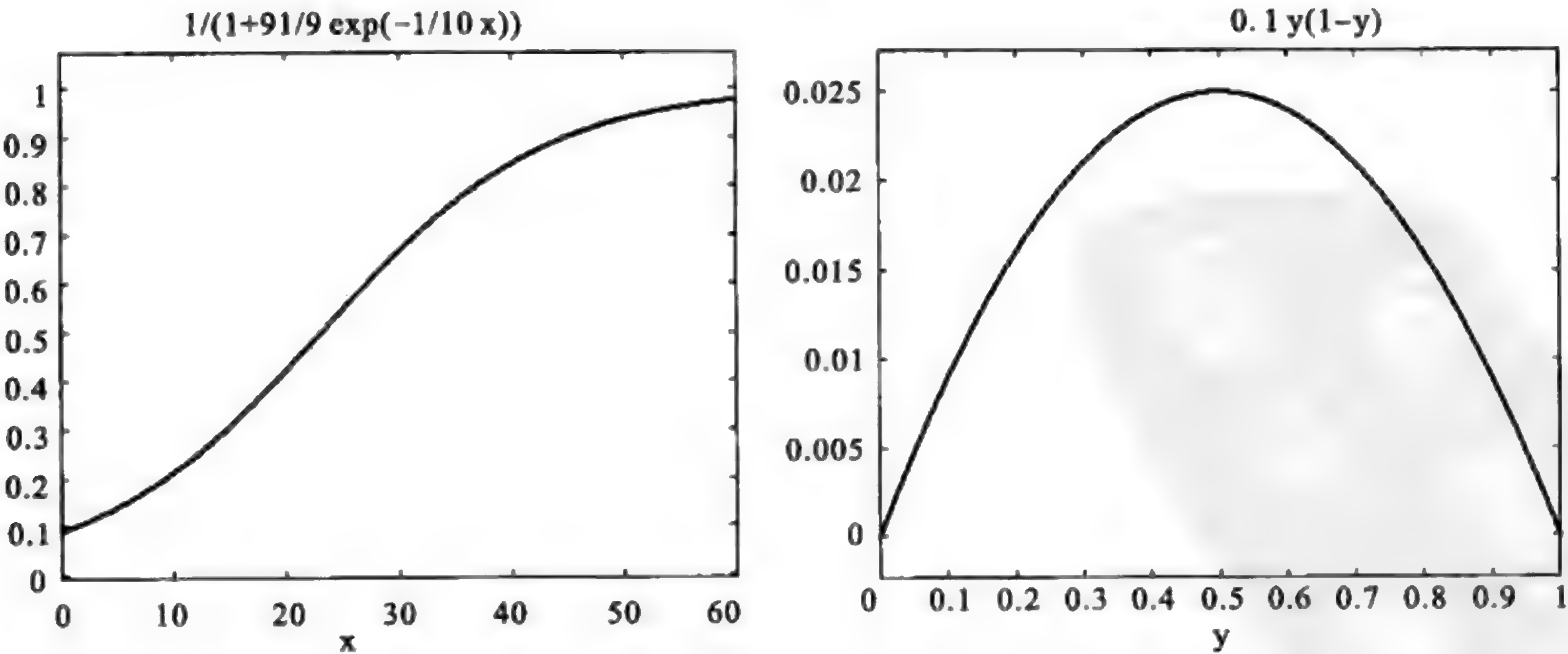


图 7.5 传染病模型 II 的函数图像及其导数的图像

这时病人增加得最快，可以认为是医院门诊量最大的一天，预示着传染病高潮的到来，是医疗卫生部门关注的时刻. t_m 与 λ 成反比，因为日接触率 λ 表示该地区的卫生水平，λ 越小卫生水平越高. 所以改善保健设施，提高卫生水平可以推迟传染病高潮的到来.

当 $t\to\infty$ 时，$i\to 1$. 即所有人终将被传染，全变为病人，这显然不符合实际情况. 其原因是模型中没有考虑到病人可以治愈，人群中的健康者只能变成病人，病人不会再变成健康者.

为了修正上述结果必须重新考虑模型的假设，在下面的两个模型中将讨论病人可以治愈的情况.

3）模型 III

有些传染病如伤风、痢疾等愈后免疫力很低，可以假定无免疫性. 于是病人被治愈后变成健康者，健康者还可以被感染再变成病人.

模型假设　模型 III 的前三个假设与模型 II 的假设相同.

(4) 每天被治愈的病人数占病人总数的比例为常数 μ，称为日治愈率. 病人治愈后成为仍可被感染的健康者，显然 $1/\mu$ 是这种传染病的平均传染期.

考虑到假设(4)，模型 II 中的(7.1.12)式应修改为

$$N\frac{\mathrm{d}i}{\mathrm{d}t}=\lambda Ns(t)i(t)-\mu Ni(t) \tag{7.1.15}$$

由于 $s(t)+i(t)=1$，所以(7.1.15)式化为

$$\frac{\mathrm{d}i}{\mathrm{d}t}=\lambda i(t)[1-i(t)]-\mu i(t),\qquad i(0)=i_0 \tag{7.1.16}$$

方程(7.1.16)的解为

$$i(t)=\begin{cases}\left[\dfrac{\lambda}{\lambda-\mu}+\left(\dfrac{1}{i_0}-\dfrac{\lambda}{\lambda-\mu}\right)\mathrm{e}^{-(\lambda-\mu)t}\right]^{-1}, & \lambda\neq\mu\\ \left(\lambda t+\dfrac{1}{i_0}\right)^{-1}, & \lambda=\mu\end{cases} \tag{7.1.17}$$

定义 $\sigma=\lambda/\mu$. 注意到 λ 和 $1/\mu$ 的含义，可知 σ 是整个传染期内每个病人有效接触的平均人数，称为接触数. 利用 σ，(7.1.16)式可改写为

$$\frac{\mathrm{d}i}{\mathrm{d}t}=-\lambda i(t)\left[i(t)-\left(1-\frac{1}{\sigma}\right)\right] \tag{7.1.18}$$

方程(7.1.18)和模型(7.1.17)的图形如图 7.6 所示.

编写 MATLAB 程序如下：

```
y=dsolve('Dy=0.01*y*(1-y)-0.05*y','y(0)=0.7','x');  % 求解此微分方程
ezplot(y,[0,120])
y2=dsolve('Dy=0.3*y*(1-y)-0.15*y','y(0)=0.7','x');  %i0=0.7,λ=0.3,σ=2
y3=dsolve('Dy=0.3*y*(1-y)-0.15*y','y(0)=0.3','x');
```

```
figure,ezplot(y2,[0,25]);
figure,ezplot(y3,[0,25])
```

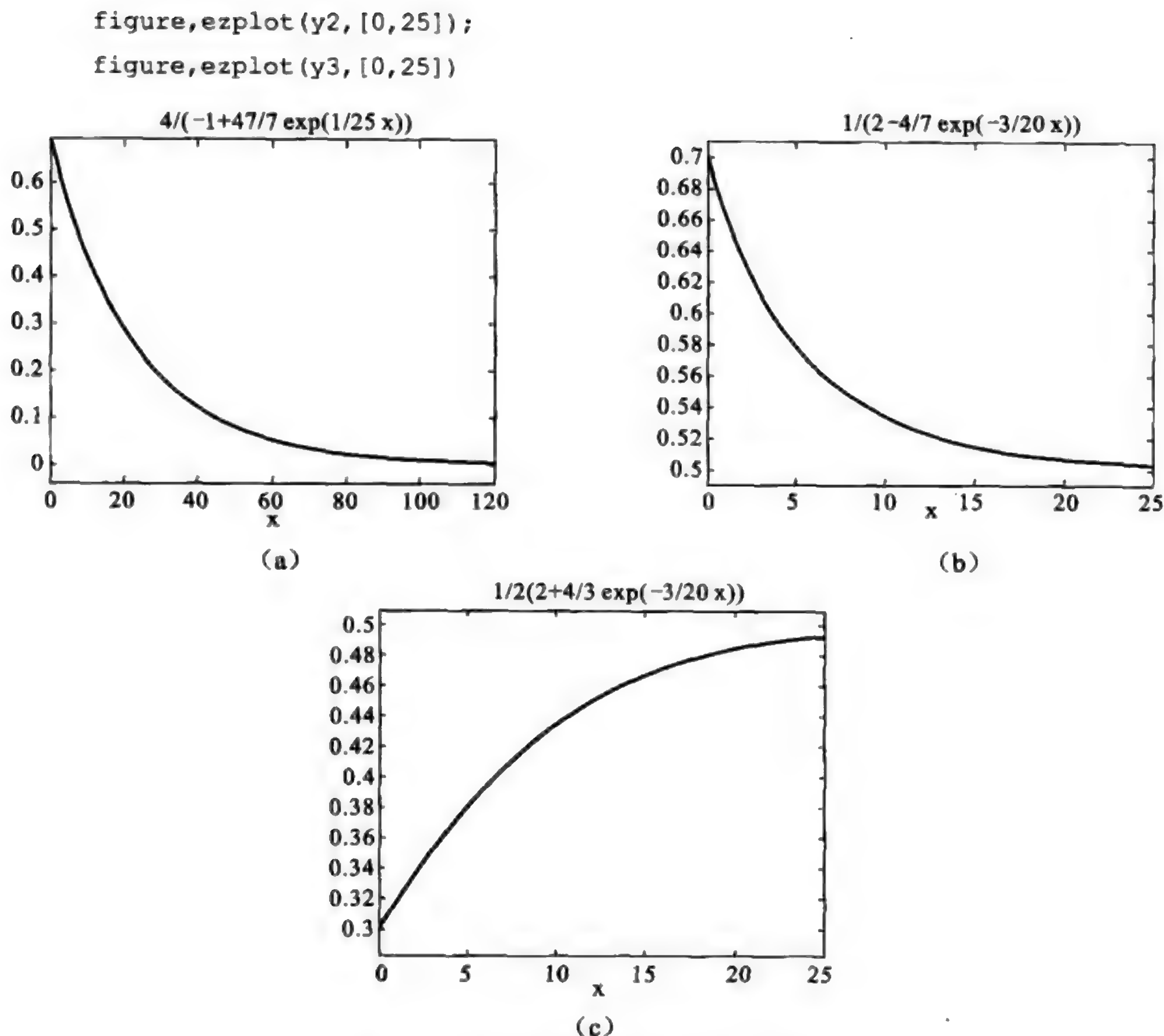


图 7.6　传染病模型 III 的增长曲线图

图 7.6 中图(a)表示 $i_0=0.7,\lambda=0.01,\sigma=0.2$ 时的传染病增长曲线图；图(b)表示 $i_0=0.7,\lambda=0.3,\sigma=2$ 时的增长曲线图；图(c)表示 $i_0=0.3,\lambda=0.3,\sigma=2$ 时的增长曲线图.

由图 7.6 不难看出，接触数 $\sigma=1$ 是一个阈值. 当 $\sigma>1$ 时，$i(t)$ 的增减性取决于 i_0 的大小，但其极限值 $i(+\infty)=1-\dfrac{1}{\sigma}$ 随着 σ 的增加而增加；当 $\sigma\leqslant 1$ 时，病人比例 $i(t)$ 越来越小，最终趋于零，这是由于传染期内健康者变成病人的人数不超过原来病人数的缘故.

4) 模型 IV

大多数传染病如天花、流感、肝炎、麻疹等治愈后均有很强的免疫力，所以病愈的人既非健康者(易感染者)，也非病人(已感染者)，他们已经退出传染系统. 这种情况比较复杂，下面将详细分析建模过程.

模型假设

(1) 在疾病传播期内所考察地区的总人数 N 不变，既不考虑生死，也不考虑迁移. 人群分为健康者、病人和病愈免疫的移出者，三类人在总人数 N 中占的比例分别记为 $s(t)$，$i(t)$ 和 $r(t)$.

(2) 病人的日接触率为常数 λ，日治愈率为常数 μ，传染期接触数为 $\sigma=\lambda/\mu$.

由假设(1)知

$$s(t)+i(t)+r(t)=1 \tag{7.1.19}$$

由假设(2)知方程(7.1.15)仍成立. 对于病愈免疫的移出者有

$$N\frac{\mathrm{d}r}{\mathrm{d}t}=\mu Ni(t) \tag{7.1.20}$$

再记初始时刻的健康者和病人的比例分别是 $s_0(s_0>0)$ 和 $i_0(i_0>0)$，且不妨假设移出者的初始值 $r_0=0$，则由(7.1.15)、(7.1.19)和(7.1.20)式得

$$\begin{cases}\dfrac{\mathrm{d}i}{\mathrm{d}t}=\lambda s(t)i(t)-\mu i(t), & i(0)=t_0\\[2mm] \dfrac{\mathrm{d}s}{\mathrm{d}t}=-\lambda s(t)i(t), & s(0)=s_0\end{cases} \tag{7.1.21}$$

(7.1.21)式即为我们所要建立的数学模型. 由于方程(7.1.21)无法求出 $s(t)$ 和 $i(t)$ 的解析解，所以我们研究其解在相平面 s-i 上的性质. 由 $\sigma=\lambda/\mu$ 消去 $\mathrm{d}t$ 得

$$\begin{cases}\dfrac{\mathrm{d}i}{\mathrm{d}s}=\dfrac{1}{\sigma s}-1\\[2mm] i|_{s=s_0}=i_0\end{cases} \tag{7.1.22}$$

具体应用时，采用数值计算，使用数学软件来完成，其 MATLAB 程序如下：

首先，建立 M-文件.

```
function y=ill(t,x)  %函数 ill,表示模型 IV
a=1;b=0.3;
y=[a*x(1)*x(2)-b*x(1),-a*x(1)*x(2)]';
```

其次，建立主程序文件

```
ts=0:50;
x0=[0.02,0.98];
[t,x]=ode45('ill',ts,x0)  %调用 ode45 求解'ill'方程组
plot(t,x(:,1),t,x(:,2)),grid,  %画出健康者和病人的变化曲线
figure,plot(x(:,2),x(:,1)),grid  %画出相图
ill.m
```

运行结果如图 7.7 所示.

由图 7.7(a)可以看出：在初始时刻健康者和病人百分比的总和为 1；病人的数量先增加然后下降，说明在某时刻传染病得到抑制；而治愈的人群退出此系统，所以最后系统的人群数量为 0；这时所有的人群均是免疫者.

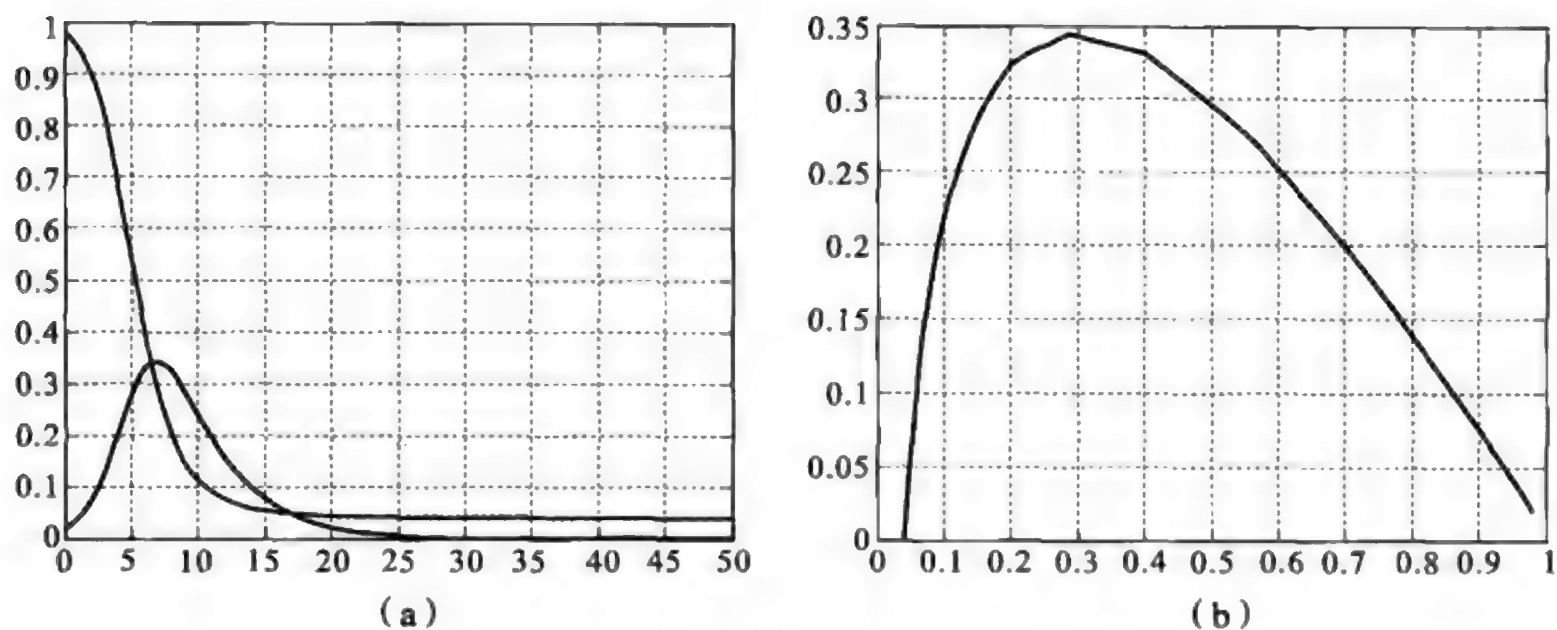


图 7.7　模型 IV 的变化曲线以及相图

关于对模型 IV 进行相轨线分析的问题，请有兴趣的同学参阅姜启源、谢金星、叶俊主编的《数学模型》.

4. Volterra 模型

意大利生物学家 D'Ancona 曾致力于鱼类种群相互制约关系的研究，在研究过程中他无意中发现了第一次世界大战期间地中海各港口捕获的几种鱼类占捕获总量百分比的资料，从这些资料中他发现各种软骨掠肉鱼，如鲨鱼、鳐鱼等，我们称之为捕食者的一些不是很理想的鱼占总渔获量的百分比，在 1914～1923 年期间，意大利阜姆港收购的捕食者所占的比例有明显的增加.

表 7.2　1914～1923 年间意大利阜姆港收购的捕食者所占的比例

年代	1914	1915	1916	1917	1918
百分比	11.9	21.4	22.1	21.2	36.4
年代	1919	1920	1921	1922	1923
百分比	27.3	16.0	15.9	14.8	10.7

模型假设

(1) 食饵由于捕食者的存在使增长率降低，假设降低的程度与捕食者数量成正比.

(2) 捕食者由于食饵为它提供食物的作用使其死亡率降低或使之增长，假定增长的程度与食饵数量成正比.

$x_1(t)$：食饵在 t 时刻的数量；

$x_2(t)$：捕食者在 t 时刻的数量；

r_1：食饵独立生存时的增长率；

r_2：捕食者独自存在时的死亡率；

λ_1:捕食者掠取食饵的能力;

λ_2:食饵对捕食者的供养能力;

e:捕获能力系数.

1) 模型 I(不考虑人工捕获)

$$\begin{cases}\dfrac{dx_1}{dt}=x_1(r_1-\lambda_1x_2)\\ \dfrac{dx_2}{dt}=x_2(-r_2+\lambda_2x_1)\end{cases}$$

该模型反映了在没有人工捕获的自然环境中食饵与捕食者之间的制约关系,没有考虑食饵和捕食者自身的阻滞作用,是 Volterra 提出的最简单的模型.

针对一组具体的数据用 MATLAB 软件进行计算.设食饵和捕食者的初始数量分别为 $x_1(0)=x_{10}$,$x_2(0)=x_{20}$ 对于数据 $r_1=1$,$\lambda_1=0.1$,$r_2=0.5$,$\lambda_2=0.02$,$x_{10}=25$,$x_{20}=2$,t 的终值经试验后确定为 15,即模型为

$$\begin{cases}x_1'=x_1(1-0.1x_2)\\ x_2'=x_2(-0.5+0.02x_1)\\ x_1(0)=25, \quad x_2(0)=2\end{cases}$$

首先,建立 M-文件 shier.m 如下:

```
function dx=shier(t,x)
dx=zeros(2,1);
dx(1)=x(1)*(1-0.1*x(2));
dx(2)=x(2)*(-0.5+0.02*x(1));
```

其次,建立主程序 shark.m 如下:

```
[t,x]=ode45('shier',[0 15],[2 52]);
plot(t,x(:,1),'-',t,x(:,2),'*')
plot(x(:,1),x(:,2))
```

求解得到数值解如图 7.8 所示.

图 7.8(a)反映了食饵和捕食者的关系.观察其图像可以猜测它们都是周期函数.

2) 模型 II(考虑人工捕获)

下面转向考虑 D'Ancona 所涉及的问题,即捕鱼业对鱼类种群的影响.考虑捕捞因素对鱼类的影响,这时反映食用鱼与非食用鱼关系的微分方程组应改写为

$$\begin{cases}x'=(a-\varepsilon)x-bxy\\ y'=-(c+\varepsilon)y+dxy\end{cases} \tag{7.1.23}$$

其中,ε 是捕捞系数,它反映了捕鱼业的捕捞强弱情况.仍取 $r_1=1$,$\lambda_1=0.1$,$r_2=0.5$,$\lambda_2=0.02$,$x_1(0)=25$,$x_2(0)=2$.设战前捕获能力系数 $e=0.3$,战争中降为 $e=0.1$,则战前与战争中的模型分别为

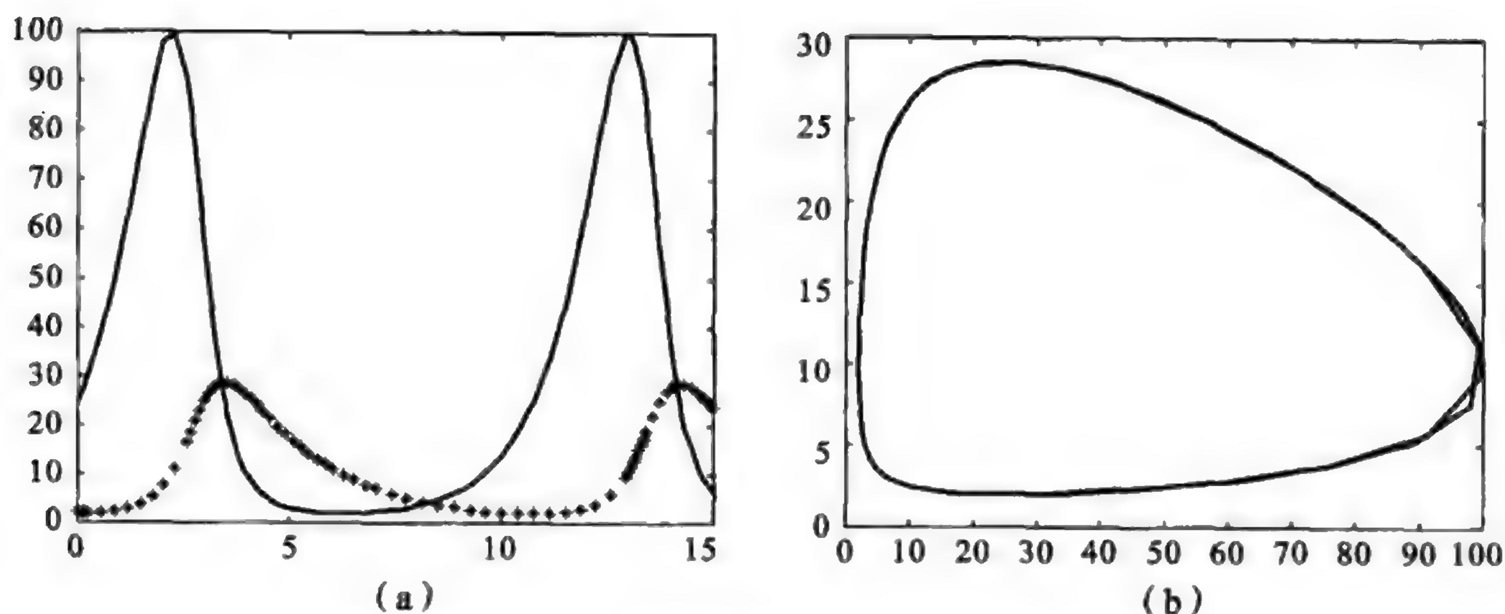


图 7.8 (捕食模型 I)食饵和捕食者的关系图及其相图

$$\begin{cases}\dfrac{dx_1}{dt}=x_1(0.7-0.1x_2)\\ \dfrac{dx_2}{dt}=x_2(-0.8+0.02x_1)\\ x_1(0)=25,\quad x_2(0)=2\end{cases}$$

$$\begin{cases}\dfrac{dx_1}{dt}=x_1(0.9-0.1x_2)\\ \dfrac{dx_2}{dt}=x_2(-0.6+0.02x_1)\\ x_1(0)=25,\quad x_2(0)=2\end{cases}$$

建立 M-文件 shier1.m：

```
function dx=shier1(t,x)
      dx=zeros(2,1);
      dx(1)=x(1)*(0.7-0.1*x(2));
      dx(2)=x(2)*(-0.8+0.02*x(1));
```

建立 M-文件 shier2.m 如下：

```
function dy=shier2(t,y)
      dy=zeros(2,1);
      dy(1)=y(1)*(0.9-0.1*y(2));
      dy(2)=y(2)*(-0.6+0.02*y(1));
```

建立主函数 shark.m 运行得到图 7.9.

```
[t1,x]=ode45('shier1',[0 15],[2 52]);
[t2,y]=ode45('shier2',[0 15],[2 52]);
x1=x(:,1);x2=x(:,2);x3=x2./(x1+x2);
y1=y(:,1);y2=y(:,2);y3=y2./(y1+y2);
plot(t1,x3,'-',t2,y3,'*')
```

图 7.9 中实线为战前的鲨鱼比例,“ * ”线为战争中的鲨鱼比例. 由图 7.9 可以得出结论:战争中鲨鱼的比例比战前高.

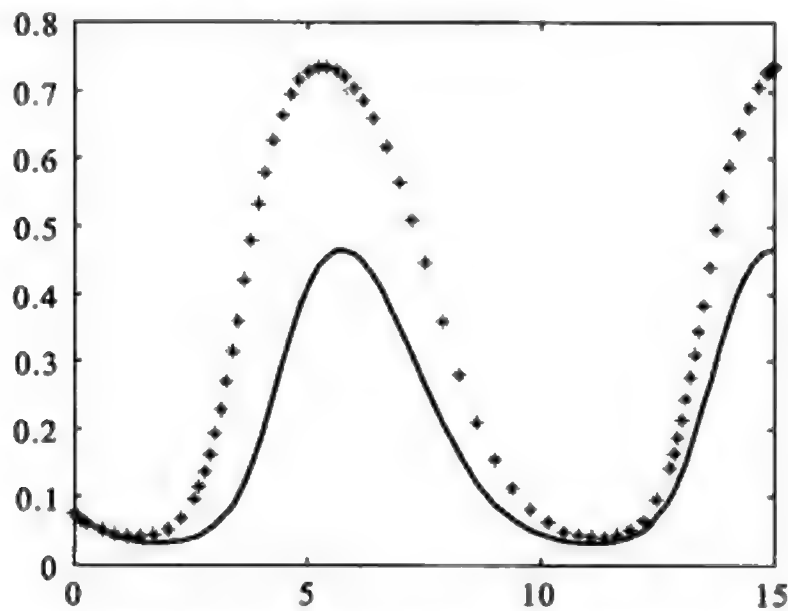


图 7.9 (捕食模型 II)食饵和捕食者的关系图及其相图

7.1.4 评论

微分方程在物理学、力学、经济学和管理科学等实际问题中具有广泛的应用. 通过本节的学习,读者可从中感受到应用数学建模的理论和方法解决实际问题的魅力.

7.2 差分方程模型

7.2.1 模型的使用背景

差分方程反映的是关于离散变量的变化规律. 针对要解决的实际问题,引入系统或过程中的离散变量,根据实际背景的规律、性质、平衡关系,建立离散变量所满足的平衡关系等式,从而建立差分方程. 通过求出和分析方程的解,或者分析得到方程解的性质(平衡性、稳定性、渐近性、振动性、周期性等),从而把握这个离散变量的变化过程的规律.

差分方程模型有着非常广泛的实际应用背景. 在经济、金融、保险、生物种群的数量结构规律分析、疾病和病虫害的控制与防治、遗传规律的研究等许多的方面都有着非常重要的作用.

7.2.2 差分方程的理论和方法

规定 t 只取非负整数. 记 y_t 为变量 y 在 t 点的取值,则称 $\Delta y_t = y_{t-1} - y_t$ 为 y_t 的一阶向前差分,简称差分,称 $\Delta^2 y_t = \Delta(\Delta y_t) = \Delta y_{t+1} - \Delta y_t = y_{t+2} - 2y_{t+1} + y_t$ 为 y_t 的二阶差分. 类似地,可以定义 y_t 的 n 阶差分 $\Delta^n y_t$.

由 t, y_t 及 y_t 的差分给出的方程称为 y_t 的差分方程，其中含 y_t 的最高阶差分的阶数称为该差分方程的阶. 差分方程也可以写成不显含差分的形式. 例如，二阶差分方程 $\Delta^2 y_t - \Delta y_t + y_t = 0$ 也可改写成 $y_{t+2} - y_{t+1} + y_t = 0$.

满足此差分方程的序列 y_t 称为差分方程的解. 类似于微分方程情况，若解中含有的独立常数的个数等于差分方程的阶数时，称此解为该差分方程的通解. 若解中不含任意常数，则称此解为满足某些初值条件的特解.

称如下形式的差分方程

$$a_0 y_{n+t} + a_1 y_{n+t-1} + \cdots + a_n y_t = b(t) \tag{7.2.1}$$

为 n 阶常系数线性差分方程，其中 $a_0, a_1, \cdots, a_n$ 是常数，$a_0 \neq 0$. 其对应的齐次方程为

$$a_0 y_{n+t} + a_1 y_{n+t-1} + \cdots + a_n y_t = 0 \tag{7.2.2}$$

容易证明，若序列 $y_t^{(1)}$ 与 $y_t^{(2)}$ 均为方程(7.2.2)的解，则 $y_t = C_1 y_t^{(1)} + C_2 y_t^{(2)}$ 也是方程(7.2.2)的解，其中 C_1, C_2 为任意常数. 若 $y_t^{(1)}$ 是方程(7.2.2)的解，$y_t^{(2)}$ 是方程(7.2.1)的解，则 $y_t = y_t^{(1)} + y_t^{(2)}$ 也是方程(7.1.1)的解.

方程(7.2.1)可用如下的代数方法求其通解：

(1) 先求解对应的特征方程：

$$a_0 \lambda^n + a_1 \lambda^{n-1} + \cdots + a_n = 0 \tag{7.2.3}$$

(2) 根据特征根的不同情况，求齐次方程(7.2.2)的通解.

① 若特征方程(7.2.3)有 n 个互不相同的实根 $\lambda_1, \lambda_2, \cdots, \lambda_n$，则齐次方程(7.2.2)的通解为

$$C_1 \lambda_1^t + C_2 \lambda_2^t + \cdots + C_n \lambda_n^t \quad (C_1, C_2, \cdots, C_n \text{ 为任意常数})$$

② 若 λ 是特征方程(7.2.3)的 k 重根，通解中对应于 λ 的项为 $(\overline{C}_1 + \cdots + \overline{C}_k t^{k-1})\lambda^t$，$\overline{C}_i (i=1,2,\cdots,k)$ 为任意常数.

③ 若特征方程(7.2.3)有单重复根 $\lambda = \alpha + \beta\mathrm{i}$，通解中对应它们的项为 $\overline{C}_1 \rho^t \cos\varphi t + \overline{C}_2 \rho^t \sin\varphi t$，其中 $\rho = \sqrt{\alpha^2 + \beta^2}$ 为 λ 的模，$\varphi = \arctan \dfrac{\beta}{\alpha}$ 为 λ 的幅角.

④ 若 $\lambda = \alpha + \beta\mathrm{i}$ 是特征方程(7.2.3)的 k 重复根，则通解对应于它们的项为

$$(\overline{C}_1 + \cdots + \overline{C}_k t^{k-1})\rho^t \cos\varphi t + (\overline{C}_{k+1} + \cdots + \overline{C}_{2k} t^{k-1})\rho^t \sin\varphi t$$

$\overline{C}_i (i=1,2,\cdots,2k)$ 为任意常数.

(3) 求非齐次方程(7.2.1)的一个特解 $\overline{y}_t$. 若 y_t 为方程(7.2.2)的通解，则非齐次方程(7.2.1)的通解为 $\overline{y}_t + y_t$.

求非齐次方程(7.2.1)的特解一般要用到常数变易法，计算较繁. 对特殊形式的 $b(t)$ 也可使用待定系数法. 例如，当 $b(t) = b^t p_k(t)$ 时，$p_k(t)$ 为 t 的 k 次多项式时可以证明：若 b 不是特征根，则非齐次方程(7.2.1)有形如 $b^t q_k(t)$ 的特解，$q_k(t)$ 也是 t 的 k 次多项式；若 b 是 r 重特征根，则方程(7.2.1)有形如 $b^t t^r q_k(t)$ 的特解. 进而可利用待定系数法求出 $q_k(t)$，从而得到方程(7.2.1)的一个特解 $\overline{y}_t$.

在应用差分方程研究问题时，我们常常需要讨论解的稳定性. 对常系数非齐次线

性差分方程(7.2.1),若不论其对应齐次方程的通解中任意常数 $C_1, C_2, \cdots, C_n$ 如何取值,当 $t \to +\infty$ 时总有 $y_t \to 0$,则称方程(7.2.1)的解是稳定的.根据通解的结构不难看出,非齐次方程(7.2.1)稳定的充要条件为其所有特征根的模均小于 1.

7.2.3 案例分析

1. 商业贷款

设现有一笔 p 万元的商业贷款,如果贷款期是 n 年,年利率是 r_1,今采用月还款的方式逐月偿还,建立数学模型计算每月的还款数是多少.

模型建立 设贷款后第 k 个月后的欠款数是 A_k 元,月还款为 m 元,月贷款利息为 $r=r_1/12$,则

$$A_k + rA_k = A_{k+1} + m$$

即

$$A_{k+1} = (1+r)A_k - m \tag{7.2.4}$$

模型求解 令 $B_k = A_k - A_{k-1}$,则 $B_k = B_{k-1}(1+r) = B_1(1+r)^{k-1}$,所以

$$\begin{aligned} A_k &= A_0 + B_1 + B_2 + \cdots + B_k \\ &= A_0 + B_1[1+(1+r)+\cdots+(1+r)^{k-1}] \\ &= A_0(1+r)^k - \frac{m}{r}[(1+r)^k - 1], \quad k=0,1,2,\cdots \end{aligned}$$

这就是差分方程(7.2.4)的解.

结果分析 将已知数据 A_0,r 代入 $A_{12}=0$ 中,可以求出月还款额 m.例如,当 $A_0=10000$,$r=0.0052125$,$n=2$ 时,可以求出 $m=444.356$ 元.

模型的进一步拓广分析:拓广分析包括条件的改变、目标的改变、某些特殊结果等.如果令 $A_k=A$,则 $A=\frac{m}{r}$,且当 $A_0=\frac{m}{r}$ 时,总有 $A_k=\frac{m}{r}$,即表明每月只还上了利息.只有当 $A_0<\frac{m}{r}$ 时,欠款余额逐步减少,并最终还上贷款.

编写 MATLAB 程序如下:

```
r=0.0052125;  %r为贷款利率
A0=10000;  %A0为贷款总额
n=2;  %n为还款年数
m=r*A0*(1+r)^(12*n)/((1+r)^(12*n)-1)  %m为月还款额
```

运行结果如下:

```
m=444.3560
```

由以上运行结果可知,月还款额为 444.3560 元.

2. 离散形式的阻滞增长模型

建立人口增长的离散形式的阻滞增长模型(logistic 模型).在不同的增长率下讨论种群数量的变化趋势.

模型及其求解　以种群繁殖的周期划分时段，记时段 k（或第 k 代）的种群数量为 x_k，增长率是 r，种群最大容量为 N，阻滞增长模型的离散形式可表为

$$x_{k+1}-x_k=r\left(1-\frac{x_k}{N}\right)x_k,\quad k=0,1,2,\cdots$$

模型分析

(1) 该差分方程有两个平衡点：$x=N$，$x=0$.

(2) $x=0$ 不稳定. $x=N$ 稳定的条件：$r<2$. 若 $r>2$，虽然这时 x_k 不收敛，但是似乎它的变化仍有某种规律.

MATLAB 演示计算：不妨设 $N=1$，初值 $x_0=0.1$. 取 $r=1.8$ 程序如下：

```
r=1.8;
x=0.1;  %赋初值
n=40;
for i=1:n
  x(i+1)=x(i)+R*x(i)*(1-r);  %迭代计算
end
k=(0:40)';
[k,xx]  %输出结果
plot(k,x),
```

表 7.3　离散形式的阻滞增长模型计算结果

k	x_k ($r=0.3$)	x_k ($r=1.8$)	x_k ($r=2.5$)
0	0.1000	0.1000	0.1000
1	0.1270	0.2620	0.3250
2	0.1603	0.6100	0.8734
3	0.2006	1.0382	1.1498
4	0.2487	0.9668	0.7192
5	0.3048	1.0246	1.2241
⋮	⋮	⋮	⋮
17	0.9529	1.0017	1.2250
18	0.9664	0.9986	0.5359
19	0.9761	1.0011	1.1577
20	0.9831	0.9991	0.7012
⋮	⋮	⋮	⋮
31	0.9997	1.0001	1.1577
32	0.9998	0.9999	0.7012
33	0.9998	1.0000	1.2250

续表

k	x_k (r=0.3)	x_k (r=1.8)	x_k (r=2.5)
34	0.9999	1.0000	0.5359
35	0.9999	1.0000	1.1577
36	0.9999	1.0000	0.7012
37	1.0000	1.0000	1.2250
38	1.0000	1.0000	0.5359
39	1.0000	1.0000	1.1577
40	1.0000	1.0000	0.7012

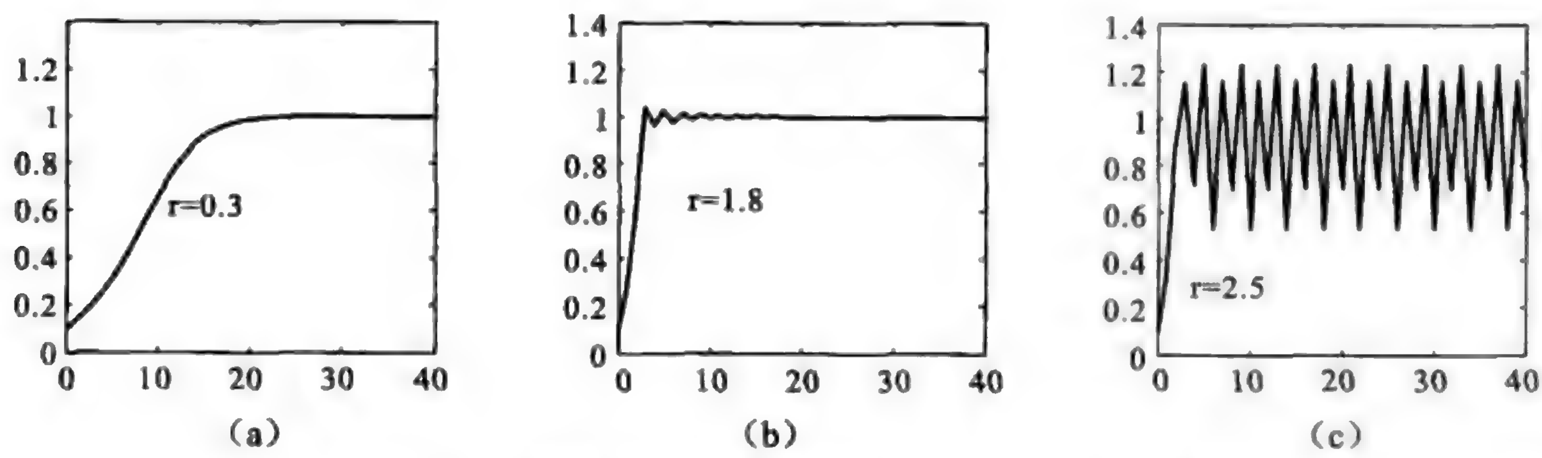


图 7.10 离散形式的阻滞增长模型 r 取不同值的图像

表 7.3 中第二列表示 r 取 0.3 时的运行结果，第三列表示 r 取 1.8 时的运行结果，第四列表示 r 取 2.5 时的运行结果. 图 7.10 分别表示了三种情况的图像. 从图 7.10 可以看出，当 r 的取值越小的时候，离散形式的解越接近连续情形下的解.

3. 汽车租赁公司的运营

一家汽车租赁公司在三个相邻的城市运营，为方便顾客，公司承诺，在一个城市租赁的汽车可以在任意一个城市归还. 根据经验估计和市场调查，一个租赁期内在 A 市租赁的汽车在 A,B,C 市归还的比例分别为 0.6,0.3,0.1；在 B 市租赁的汽车在 A,B,C 市归还的比例分别为 0.2,0.7,0.1；在 C 市租赁的汽车在 A,B,C 市归还的比例分别为 0.1,0.3,0.6. 若公司开业时将 N 辆汽车按一定方式分配到三个城市，建立运营过程中汽车数量在三个城市间转移的模型，并讨论时间充分长以后的变化趋势.

模型及其求解 记第 k 个租赁期末公司在 A,B,C 市的汽车数量分别为 $x_1(k)$，$x_2(k)$，$x_3(k)$，容易写出第 $k+1$ 个租赁期末公司在 A,B,C 市的汽车数量为($k=0,1,2,\cdots$)

$$\begin{cases} x_1(k+1)=0.6x_1(k)+0.2x_2(k)+0.1x_3(k) \\ x_2(k+1)=0.3x_1(k)+0.7x_2(k)+0.3x_3(k) \\ x_3(k+1)=0.1x_1(k)+0.1x_2(k)+0.6x_3(k) \end{cases}$$

记向量 $\boldsymbol{x}(k)=[x_1(k),x_2(k),x_3(k)]^{\mathrm{T}}$，矩阵

$$
\mathbf{A}=\begin{bmatrix}0.6 & 0.2 & 0.1\\0.3 & 0.7 & 0.3\\0.1 & 0.1 & 0.6\end{bmatrix}
$$

则

$$
\boldsymbol{x}(k+1)=\mathbf{A}\boldsymbol{x}(k),\quad k=0,1,2,\cdots
$$

给定初始值 $\boldsymbol{x}(0)$,可以计算各个租赁期三个城市汽车数量的变化.

模型分析 猜想:时间充分长以后三个城市的汽车数量趋向稳定,并且稳定值汽车的初始分配无关. 为了证实该猜想,记稳定值为 $\boldsymbol{x}$,由 $\boldsymbol{x}$ 应满足 $\mathbf{A}\boldsymbol{x}=\boldsymbol{x}$ 表明矩阵 $\mathbf{A}$ 的一个特征根 $\lambda=1$,且 $\boldsymbol{x}$ 是对应的特征向量.

MATLAB 演示计算,初始分配城市 A:200,城市 B:200,城市 C:200. 计算并作图,程序如下:

```
A=[0.6,0.2,0.1;0.3,0.7,0.3;0.1,0.1,0.6];
x(:,1)=[200,200,200]';   %赋初值
n=10;
for k=1:n
    x(:,k+1)=A*x(:,k);   %迭代计算
end
round(x),
k=0:10;
plot(k,x),grid,
```

运行结果如下:

表 7.4 三个城市的汽车数量

k	0	1	2	3	4	5	6	7	8	9	10
$x_1(k)$	200	180	176	176	178	179	179	180	180	180	180
$x_2(k)$	200	260	284	294	297	299	300	300	300	300	300
$x_3(k)$	200	160	140	130	125	123	121	121	120	120	120

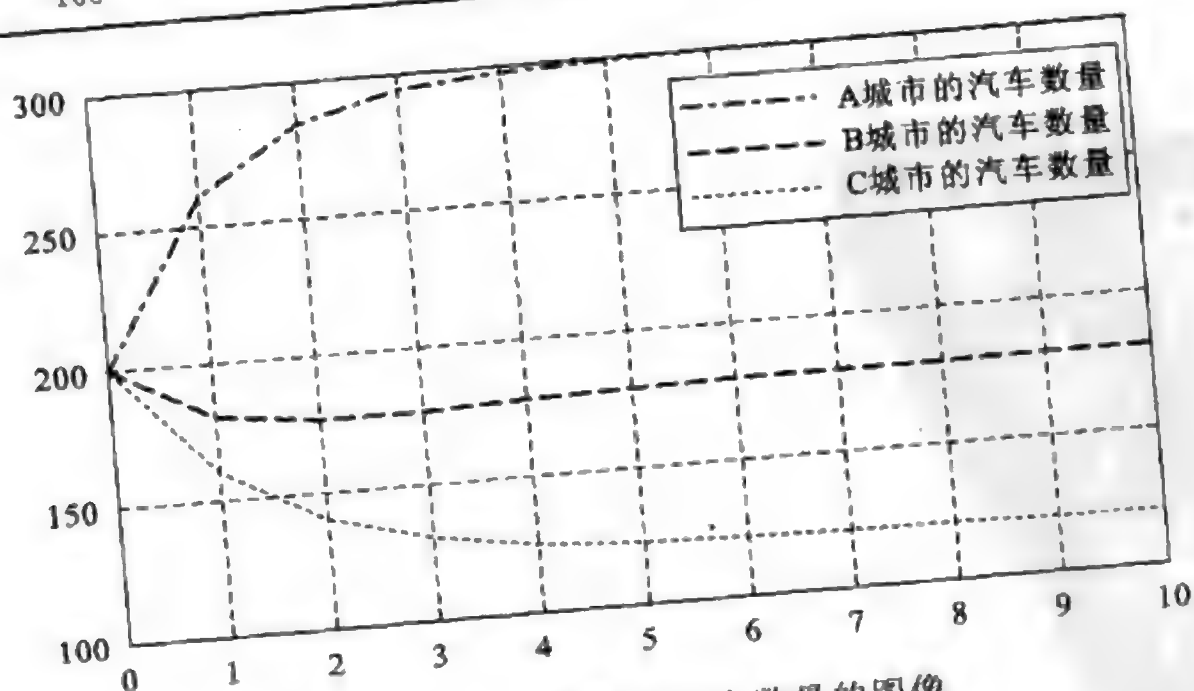


图 7.11 三个城市的汽车数量的图像

由图 7.11 可以清晰地看出，时间充分长以后三个城市的汽车数量趋向稳定，并且稳定值与汽车的初始分配无关.

4. 动物养殖问题

养殖场养殖一类动物最多三年(满三年的将送往市场卖掉)，按一岁、二岁和三岁将其分为三个年龄组. 一龄组是幼龄组，二龄组和三龄组是有繁殖后代能力的成年组. 二龄组平均一年繁殖 4 个后代，三龄组平均一年繁殖 3 个后代. 一龄组和二龄组动物能养殖成为下一年龄组动物的成功率分别为 0.5 和 0.25. 假设刚开始养殖时有三个年龄组的动物各 1000 头.

(1) 求一年后、二年后、三年后各年龄组动物数量.

(2) 五年后农场三个年龄组的动物的情况会怎样?

(3) 如果每年平均向市场供应动物数 $\boldsymbol{c}=[s,s,s]^{\mathrm{T}}$，考虑每年都必须保持有每一年龄组的动物前提下，$\boldsymbol{c}$ 应取多少为好? 是否有最佳方案?

模型建立 由题设，在初始时刻一岁、二岁、三岁的动物数量分别为

$$x_1^{(0)}=1000, \qquad x_2^{(0)}=1000, \qquad x_3^{(0)}=1000$$

以一年为一时间段，则某时刻三个年龄组的动物数量可用向量

$$\boldsymbol{x}=[x_1,x_2,x_3]^{\mathrm{T}}$$

表示. 用向量

$$\boldsymbol{x}^{(k)}=[x_1^{(k)},x_2^{(k)},x_3^{(k)}]^{\mathrm{T}}$$

表示第 k 个时间段动物数分布. 当 $k=0,1,2,3$ 时，$\boldsymbol{x}^{(k)}$ 分别表示养殖开始时、一年后、两年后、三年后的动物数量分布. 根据二龄组和三龄组动物的繁殖能力，在第 k 个时间段，二龄组动物在其年龄段平均繁殖 4 个后代，三龄组动物在其年龄段平均繁殖 3 个后代. 由此得第一个年龄组在第 $k+1$ 个时间段的数量如下：

$$x_1^{(k+1)}=4x_2^{(k)}+3x_3^{(k)}$$

同理，根据一龄组和二龄组的养殖成功率，可得等式

$$x_2^{(k+1)}=0.5x_1^{(k)}, \qquad x_3^{(k+1)}=0.25x_2^{(k)}$$

建立数学模型如下：

$$\begin{cases} x_1^{(k+1)}=4x_2^{(k)}+3x_3^{(k)}, \\ x_2^{(k+1)}=0.5x_1^{(k)}, \\ x_3^{(k+1)}=0.25x_2^{(k)}, \end{cases} \qquad k=0,1,2,3 \tag{7.2.5}$$

或写成矩阵形式：

$$\begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ x_3^{(k+1)} \end{bmatrix}=\begin{bmatrix} 0 & 4 & 3 \\ 0.5 & 0 & 0 \\ 0 & 0.25 & 0 \end{bmatrix}\begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ x_3^{(k)} \end{bmatrix}, \quad k=0,1,2,3 \tag{7.2.6}$$

由此得向量 $\boldsymbol{x}^{(k)}$ 和 $\boldsymbol{x}^{(k+1)}$ 的递推关系式：

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{L}\boldsymbol{x}^{(k)} \tag{7.2.7}$$

其中,矩阵

$$
L=\begin{bmatrix}0 & 4 & 3\\ 0.5 & 0 & 0\\ 0 & 0.25 & 0\end{bmatrix}
$$

称为莱斯利矩阵. 由(7.2.7)式可得

$$
x^{(k+1)}=L^{k+1}x^{(0)}
$$

编写 MATLAB 程序如下:

(1) 由初始数据计算一年后、两年后、三年后动物数量.

```
x0=[1000;1000;1000];
L=[0 4 3;1/2 0 0;0 1/4 0];
x1=L*x0;
x2=L*x1;
x3=L*x2;
[x1';x2';x3']
x5=L*L*x3;
```

运行结果如下:

```
ans=
        7000         500         250
        2750        3500         125
       14375        1375         875
ans=  1.0e+004*
          2.9781    0.4063  0.1797
```

(2) 计算五年内动物数量变化规律.

```
x0=[1000;1000;1000];
L=[0 4 3;1/2 0 0;0 1/4 0];
X=x0;
x(1)=X(1);y(1)=X(2);z=X(3);
for k=2:6
    X=L*X;
    x(k)=X(1);y(k)=X(2);z(k)=X(3);
end
t=0:5;
bar(t,x),
figure,bar(t,y)
figure,bar(t,z)
```

运行可以得到如图 7.12 所示三个图像,分别表示三龄组动物数量在五年内发展变化规律.

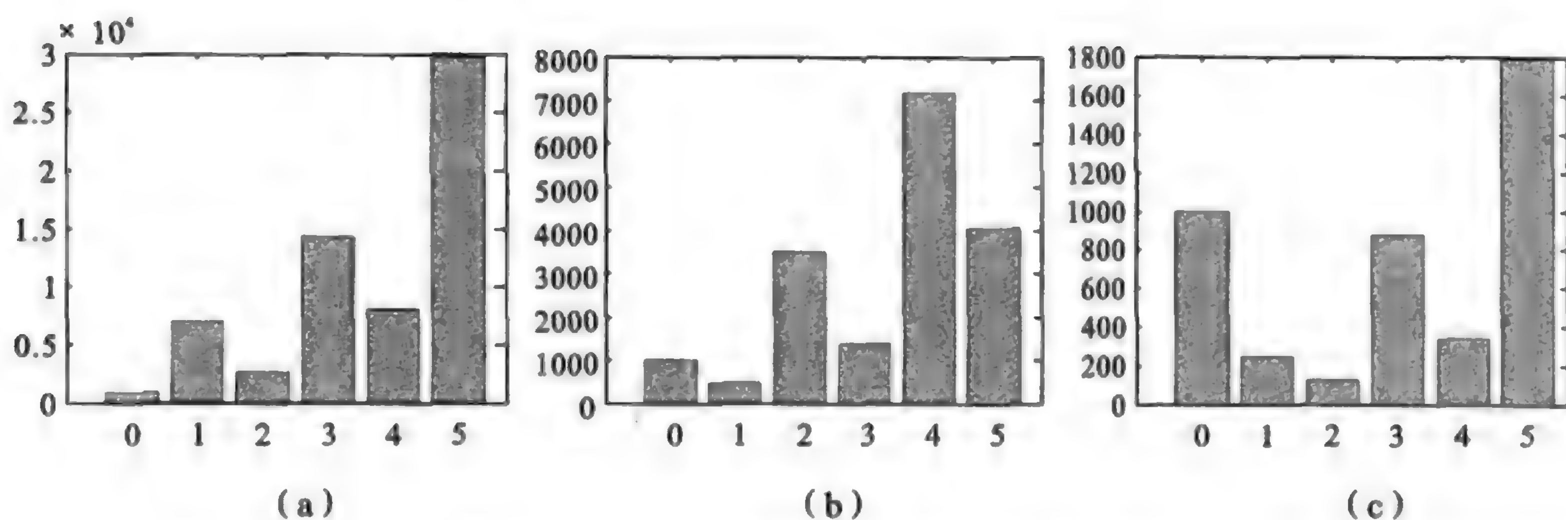


图 7.12　三龄组动物五年数量变化直方图

其中,图 7.12(a)、(b)、(c)分别表示一岁、二岁、三岁的动物数量在五年内的变化情况.

(3) 如果每年平均向市场出售动物 $\boldsymbol{c}=[s,s,s]^{\mathrm{T}}$,分析动物数分布向量变化规律可知

$$\boldsymbol{x}(1)=\boldsymbol{A}\boldsymbol{x}(0)-\boldsymbol{c}$$
$$\boldsymbol{x}(2)=\boldsymbol{A}\boldsymbol{x}(1)-\boldsymbol{c}$$
$$\boldsymbol{x}(3)=\boldsymbol{A}\boldsymbol{x}(2)-\boldsymbol{c}$$
$$\boldsymbol{x}(4)=\boldsymbol{A}\boldsymbol{x}(3)-\boldsymbol{c}$$
$$\boldsymbol{x}(5)=\boldsymbol{A}\boldsymbol{x}(4)-\boldsymbol{c}$$

所以有
$$\boldsymbol{x}(5)=\boldsymbol{A}^5\boldsymbol{x}(0)-(\boldsymbol{A}^4+\boldsymbol{A}^3+\boldsymbol{A}^2+\boldsymbol{A}+\boldsymbol{I})\boldsymbol{c}$$

考虑每年都必须保持有每一年龄的动物,应有 $\boldsymbol{x}(k)>0\ (k=1,2,3,4,5)$.

MATLAB 程序如下:

```
c=input('inputc:=');
x0=[1000;1000;1000];
L=[0 4 3;1/2 0 0;0 1/4 0];
x1=L*x0-c;
x2=L*x1-c;
x3=L*x2-c;
x4=L*x3-c;
x5=L*x4-c;
disp([x1';x2';x3';x4';x5']);
```

程序运行时输入不同的参数 c,观察数据计算结果.由实验结果可知,当取 c=100 时,能保证每一年龄动物数量不为零.

第8章 模糊数学

8.1 模糊模式识别

模式识别属于判别一个对象属于哪个模式的问题.进行模式识别要求具备两个特征:一是事先已知若干标准模式,称为标准模式库;二是有待识别的对象.所谓模糊模式识别,是指在模式识别中,模式是模糊的,或说标准模式库中提供的模式是模糊的.该方法可用于诸如汽车牌照识别、空气污染等级识别、土壤质量识别及生态环境识别等问题.

8.1.1 理论介绍

进行模糊模式识别时,要用到以下原则:

1. 最大隶属原则

最大隶属原则Ⅰ 设 $\boldsymbol{A}_1,\boldsymbol{A}_2,\cdots,\boldsymbol{A}_m$ 为给定论域 U 上的 m 个模糊模式,$x_0\in U$ 为一个待识别对象,若 $\boldsymbol{A}_i(x_0)=\max\{\boldsymbol{A}_1(x_0),\boldsymbol{A}_2(x_0),\cdots,\boldsymbol{A}_m(x_0)\}$,则认为 x_0 优先属于模糊模式 $\boldsymbol{A}_i$.

最大隶属原则Ⅱ 设 $\boldsymbol{A}$ 为给定论域 U 上的一个模糊模式,$x_1,x_2,\cdots,x_n$ 为 U 中的 n 个待识别对象,若 $\boldsymbol{A}(x_i)=\max\{\boldsymbol{A}(x_1),\boldsymbol{A}(x_2),\cdots,\boldsymbol{A}(x_n)\}$,则认为 x_i 优先属于模糊模式 $\boldsymbol{A}$.

2. 择近原则

1) 贴近度

贴近度即两个模糊集 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 之间贴近的程度,通常用 $\sigma(\boldsymbol{A},\boldsymbol{B})$ 来表达.贴近度越大,表示两个模糊集越接近.常用的贴近度有以下几种:

(1) 格贴近度:

$$\sigma_0(\boldsymbol{A},\boldsymbol{B})=\frac{1}{2}[\boldsymbol{A}\circ\boldsymbol{B}+(1-\boldsymbol{A}\odot\boldsymbol{B})]$$

其中,$\boldsymbol{A}\circ\boldsymbol{B}=\max\{\boldsymbol{A}(x)\wedge\boldsymbol{B}(x)\}$,$\boldsymbol{A}\odot\boldsymbol{B}=\min\{\boldsymbol{A}(x)\vee\boldsymbol{B}(x)\}$.

(2) 最小最大贴近度:

$$\sigma_1(\boldsymbol{A},\boldsymbol{B})=\frac{\sum_{k=1}^{n}[\boldsymbol{A}(x_k)\wedge\boldsymbol{B}(x_k)]}{\sum_{k=1}^{n}[\boldsymbol{A}(x_k)\vee\boldsymbol{B}(x_k)]}$$

(3) 最小平均贴近度：

$$\sigma_2(\boldsymbol{A},\boldsymbol{B}) = \frac{2\sum_{k=1}^{n}[\boldsymbol{A}(x_k)\wedge\boldsymbol{B}(x_k)]}{\sum_{k=1}^{n}[\boldsymbol{A}(x_k)+\boldsymbol{B}(x_k)]}$$

(4) 海明贴近度：

$$\sigma_3(\boldsymbol{A},\boldsymbol{B}) = 1-\frac{1}{n}\sum_{k=1}^{n}\left|\boldsymbol{A}(x_k)-\boldsymbol{B}(x_k)\right|$$

(5) 欧几里得贴近度：

$$\sigma_4(\boldsymbol{A},\boldsymbol{B}) = 1-\frac{1}{\sqrt{n}}\left\{\sum_{k=1}^{n}[\boldsymbol{A}(x_k)-\boldsymbol{B}(x_k)]^2\right\}^{1/2}$$

2) 择近原则

设 $U=\{x_1,x_2,x_3,x_4,x_5\}$为论域 U 上的 n 个模糊模式，$\boldsymbol{B}$ 为 U 上的一个待识别对象，若 $\sigma(\boldsymbol{B},\boldsymbol{A}_i)=\max\{\sigma(\boldsymbol{B},\boldsymbol{A}_1),\sigma(\boldsymbol{B},\boldsymbol{A}_2),\cdots,\sigma(\boldsymbol{B},\boldsymbol{A}_n)\}$，则应认为 $\boldsymbol{B}$ 应优先归属于模式 $\boldsymbol{A}_i$.

8.1.2 案例分析及编程

例 8.1 设论域 $U=\{x_1,x_2,x_3,x_4,x_5\}$上的三个模糊模式分别为 $\boldsymbol{A}=(0.7,0.5,0.2,0.4)$，$\boldsymbol{B}=(0.8,0.3,0.4,0.2)$，$\boldsymbol{C}=(0.6,0.4,0.5,0.3)$，判别 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 中哪个和 $\boldsymbol{C}$ 最贴近.

分析 按照格贴近度，有如下计算：

$$\boldsymbol{A}\circ\boldsymbol{C}=0.6\vee0.4\vee0.2\vee0.3=0.6$$
$$\boldsymbol{A}\odot\boldsymbol{C}=0.7\wedge0.5\wedge0.5\wedge0.4=0.4$$
$$\boldsymbol{B}\circ\boldsymbol{C}=0.6\vee0.3\vee0.4\vee0.2=0.6$$
$$\boldsymbol{B}\odot\boldsymbol{C}=0.8\wedge0.4\wedge0.4\wedge0.3=0.3$$
$$\sigma_1(\boldsymbol{A},\boldsymbol{C})=\frac{1}{2}[0.6+(1-0.4)]=0.6$$
$$\sigma_2(\boldsymbol{B},\boldsymbol{C})=\frac{1}{2}[0.6+(1-0.3)]=0.65$$

比较贴近度知，$\boldsymbol{B}$ 比 $\boldsymbol{A}$ 更贴近于 $\boldsymbol{C}$.

建立 M- 函数：

```
function[C]=fuzzy_mssb(model,A,B)   %模糊模式识别
C=[];
[m,n]=size(A);  %A中按行容纳待识别模式
s=length(B);  %B是标准模式
if(n~=s)
```

```
        disp('两个向量的维数不一致');
    else
        if(model==1)  %格贴近度
            for(i=1:m)
                x=max(min(A(i,:),B));
                y=min(max(A(i,:),B));
                C(1,i)=(x+(1-y))/2;
            end
        elseif(model==2)  %最小最大贴近度
            for(i=1:m)
                x=sum(min(A(i,:),B));
                y=sum(max(A(i,:),B));
                C(1,i)=x/y;
            end
        elseif(model==3)  %最小平均贴近度
            for(i=1:m)
                x=2*sum(min(A(i,:),B));
                y=sum(A(i,:)+B);
                C(1,i)=x/y;
            end
        elseif(model==4)  %海明贴近度
            for(i=1:m)
                C(1,i)=1-sum(abs(A(i,:)-B))/n;
            end
        elseif(model==5)  %欧几里得贴近度
            for(i=1:m)
                C(1,i)=1-sqrt((A(i,:)-B)*(A(i,:)-B)')/sqrt(n);
            end
        else
            disp('模型赋值不当');
        end
    end
```

输入及运行结果：

```
%输入原始数据
A=[0.7 0.5 0.2 0.4;0.8 0.3 0.4 0.2];  %模式 A 和 B
B=[0.6 0.4 0.5 0.3];  %模式 C
fuzzy_mssb(1,A,B)  %调用函数
```

```
ans=

    0.6000    0.6500
```

例 8.2 设 $U=\{x_1,x_2,x_3,x_4,x_5,x_6\}$，$\boldsymbol{A}_1,\boldsymbol{A}_2,\boldsymbol{A}_3,\boldsymbol{A}_4,\boldsymbol{A}_5,\boldsymbol{A}_6$ 为 U 上 6 个模糊模式，且 $\boldsymbol{A}_1=(1,0.8,0.5,0.4,0,0.1)$，$\boldsymbol{A}_2=(0.5,0.1,0.8,1,0.6,0)$，$\boldsymbol{A}_3=(0,1,0.2,0.7,0.5,0.8)$，$\boldsymbol{A}_4=(0.4,0,1,0.9,0.6,0.5)$，$\boldsymbol{A}_5=(0.8,0.2,0,0.5,1,0.7)$，$\boldsymbol{A}_6=(0.5,0.7,0.8,0,0.5,1)$. 现给定一个待识别对象 $\boldsymbol{B}=(0.7,0.2,0.1,0.4,1,0.8)$，试判别 B 归属于哪个模式比较合理.

分析 采用最小最大贴近度公式计算 $\boldsymbol{B}$ 与 $\boldsymbol{A}_i$ 的贴近度如下：

$\sigma(\boldsymbol{B},\boldsymbol{A}_1)=0.3333$，　$\sigma(\boldsymbol{B},\boldsymbol{A}_2)=0.3778$，　$\sigma(\boldsymbol{B},\boldsymbol{A}_3)=0.4545$

$\sigma(\boldsymbol{B},\boldsymbol{A}_4)=0.4348$，　$\sigma(\boldsymbol{B},\boldsymbol{A}_5)=0.8824$，　$\sigma(\boldsymbol{B},\boldsymbol{A}_6)=0.4565$

由于 $\sigma(\boldsymbol{B},\boldsymbol{A}_5)=\max\{\sigma(\boldsymbol{B},\boldsymbol{A}_i)\mid i=1,2,3,4,5,6\}$，由择近原则 $\boldsymbol{B}$ 应优先归属于模式 $\boldsymbol{A}_5$.

M 文件同例 8.1.

输入及运行结果：

```
A=[1 0.8 0.5 0.4 0 0.1;0.5 0.1 0.8 1 0.6 0;0 1 0.2 0.7 0.5 0.8;
   0.4 0 1 0.9 0.6 0.5;0.8 0.2 0 0.5 1 0.7;0.5 0.7 0.8 0 0.5 1];  %6个标准模式
B=[0.70.20.10.410.8];  %待判别对象
fuzzy_mssb(2,A,B)  %调用函数使用最大最小贴近度判别
ans=
    0.3333    0.3778    0.4545    0.4348    0.8824    0.4565
```

例 8.3 表 8.1 是土壤中重金属污染等级标准，表 8.2 是三个样品中重金属含量，试根据标准确定三个样品重金属污染程度.

表 8.1　土壤重金属污染等级标准　　单位:mg

级别	清洁	尚清洁	轻污染	中污染	重污染
砷	10	17	30	50	70
铅	23.35	36.09	150	350	500
镉	0.1204	0.2523	0.6	1.4	2
铬	74.88	99.54	150	350	500
汞	0.092	0.2592	0.45	1.05	1.5

表 8.2　样品中各重金属含量　　单位:mg

名称	砷	铅	镉	铬	汞
1	14.0	67.3	0.12	59.9	0.29
2	4.33	37.1	0.21	53.6	0.15
3	5.95	46.5	0.23	56.1	0.12

分析 为了用模糊模式识别来处理上述重金属污染等级问题，需要把标准等级视为标准模式，样品视为待识别模式，同时还要把它们转化为模糊模式，为此要引入隶属函数. 而隶属函数的建立通常跟指标有关. 一般来说，有的指标值是越大越好，有的指标值是越小越好，为此我们可分别采用如下形式的隶属函数：

$$A(x)=\begin{cases}0, & x\leqslant a,\\ 1-e^{-(k-a)^2}, & x>a(k>0),\end{cases}\qquad A(x)=\begin{cases}1, & x\leqslant a,\\ e^{-(k-a)^2}, & x>a(k>0)\end{cases}$$

由于本题中各指标均是值越小污染状况越轻，因而可用后式进行转化. 取 $k=1$，通过计算，土壤重金属污染等级标准各指标值可转化为以下隶属度：

1.0000	1.0000	1.0000	1.0000	1.0000
0.0000	0.0000	0.9828	0.0000	0.9724
0.0000	0	0.7945	0	0.8797
0	0	0.1945	0	0.3994
0	0	0.0292	0	0.1377

样品中各重金属含量转化为以下隶属度：

0.0000	0	1.0000	1.0000	0.9616
1.0000	0.0000	0.9920	1.0000	0.9966
1.0000	0.0000	0.9881	1.0000	0.9992

采用最小平均贴近度计算三个样品与各污染等级间的贴近度，得样品一与各污染等级间的贴近度分别为

0.7440　0.7909　0.7223　0.3341　0.1067

样品二与各污染等级间的贴近度分别为

0.8875　0.6579　0.5913　0.2592　0.0803

样品三与各污染等级间的贴近度分别为

0.8873　0.6580　0.5914　0.2593　0.0804

根据最大隶属度原则，可得三个样品重金属污染等级分别为尚清洁、清洁、清洁.

(1) 建立将标准模式转化为模糊模式的 M- 函数.

```
function[C]=fuzzy_lshsA(cs,k,A)  %将 A 用隶属度表达出来
[m,n]=size(A);
C=min(A);
if(cs==1)  %适用于指标越小越好的情形
    for(i=1:m)
        for(j=1:n)
            if(A(i,j)<=C(j))
              A(i,j)=1;
          else
              A(i,j)=exp(-k*(A(i,j)-C(j))^2);
```

```
            end
        end
    end
else  %适用于指标越大越好的情形
    for(i=1:m)
        for(j=1:n)
            if(A(i,j)<=C(j))
              A(i,j)=0;
          else
              A(i,j)=1-exp(-k*(A(i,j)-C(j))^2);
          end
        end
    end
end
```

(2) 建立将待识别模式转化为模糊模式的 M-函数.

```
function[D]=fuzzy_lshsB(cs,k,A,B)  %将B用隶属度表达出来
[m,n]=size(A);
[p,q]=size(B);
C=min(A);
if(cs==1)  %适用于指标越小越好的情形
    for(i=1:p)
        for(j=1:n)
            if(B(i,j)<=C(j))
              B(i,j)=1;
          else
              B(i,j)=exp(-k*(B(i,j)-C(j))^2);
          end
        end
    end
else  %适用于指标越大越好的情形
    for(i=1:p)
        for(j=1:n)
            if(B(i,j)<=C(j))
              B(i,j)=0;
          else
              B(i,j)=1-exp(-k*(B(i,j)-C(j))^2);
          end
        end
```

```
  end
end
```

输入及结果：

```
A=[10 23.35 0.1204 74.88 0.092;  %重金属污染等级标准
   17 36.09 0.2523 99.54 0.2592;
   30 1500.6 150 0.45;
   50 350 1.4 350 1.05;
   70 500 2500 1.5];
B= [14 67.3 0.12 59.9 0.29;  %待测样品
    4.33 37.1 0.21 5 3.6 0.15
    5.95 46.5 0.23 56.1 0.12];
    [C]=fuzzy_lshsA(1,1,A)
    [D]=fuzzy_lshsB(1,1,A,B)
```

程序运行结果为两部分，分别为上述解题过程中土壤重金属污染等级标准各指标值及样品中各重金属含量转化为隶属度后的两个矩阵. 接下来再键入语句：

```
fuzzy_mssb(3,C,D(1,:))
fuzzy_mssb(3,C,D(2,:))
fuzzy_mssb(3,C,D(3,:))  %分别做 A 的隶属度与各样品的隶属度的模式识别
```

得各样品与各等级间的贴近度：

0.7440	0.7909	0.7223	0.3341	0.1067
0.8875	0.6579	0.5913	0.2592	0.0803
0.8873	0.6580	0.5914	0.2593	0.0804

8.1.3 方法评论

M-函数的运用同时把几种贴近度都包容其中，具体到某种方法，只需作相应参数的调整. 在处理实际问题时，在命令窗口键入原始数据矩阵及含参函数，也显得非常简洁. 要注意的是，在调用函数时，当前路径必须指向 M-函数所在的目录. 而在将标准模式及待识别模式转化为模糊模式时，要学会构造合理的隶属函数. 前面给出的贴近度，可灵活选择，当某种贴近度在判别失效，如结果出现两个贴近度相等时，可尝试换另一种贴近度. 一般来说，各种贴近度判别的结果基本一致，但是由于不同人在构造贴近度函数时不同的主观偏向，可能在个别模式的判别上存在出入.

8.2 模糊综合评判

在对许多事物进行客观评判时，其评判因素往往很多，我们不能只根据某一个指标的好坏就作出判断，而应该依据多种因素进行综合评判，如技术方案的选择、

经济发展的比较等. 模糊综合评判可有效地对受多种因素影响的事物作出全面评价.

8.2.1 理论介绍

模糊综合评判通常包括以下三个方面：设与被评价事物相关的因素有 n 个，记为 $U=\{u_1,u_2,\cdots,u_n\}$，称之为因素集. 又设所有可能出现的评语有 m 个，记为 $V=\{v_1,v_2,\cdots,v_m\}$，称之为评判集. 由于各种因素所处地位不同，作用也不一样，通常考虑用权重来衡量，记为 $\boldsymbol{A}=\{a_1,a_2,\cdots,a_n\}$.

1. 评判步骤

进行模糊综合评判通常按以下步骤进行：

(1) 确定因素集 $U=\{u_1,u_2,\cdots,u_n\}$.

(2) 确定评判集 $V=\{v_1,v_2,\cdots,v_m\}$.

(3) 进行单因素评判得 $r_i=\{v_{i1},v_{i2},\cdots,v_{im}\}$.

(4) 构造综合评判矩阵：

$$\boldsymbol{R}=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \vdots & \vdots & & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nm} \end{bmatrix}$$

(5) 综合评判：对于权重 $\boldsymbol{A}=\{a_1,a_2,\cdots,a_n\}$，计算 $\boldsymbol{B}=\boldsymbol{A}\circ\boldsymbol{R}$，并根据最大隶属度原则作出评判.

2. 算子$\circ$ 的定义

在进行综合评判时，根据算子$\circ$ 的不同定义，可以得到不同的模型.

1) 模型 I：$M(\wedge,\vee)$—— 主因素决定型

运算法则为 $b_j=\max\{(a_i\wedge r_{ij}),i=1,2,\cdots,n\}\ (j=1,2,\cdots,m)$. 该模型评判结果只取决于在总评判中起主要作用的那个因素，其余因素均不影响评判结果，比较适用于单项评判最优就能认为综合评判最优的情形.

2) 模型 II：$M(\cdot,\vee)$—— 主因素突出型

运算法则为 $b_j=\max\{(a_i\cdot r_{ij}),i=1,2,\cdots,n\}\ (j=1,2,\cdots,m)$. 该模型与模型 I 比较相近，但比模型 I 精细些，不仅突出了主要因素，也兼顾了其他因素，比较适用于模型 I 失效，即不可区别而需要加细时的情形.

3) 模型 III：$M(\cdot,+)$—— 加权平均型

运算法则为 $b_j=\sum_{i=1}^{n}a_i\cdot r_{ij}\,(j=1,2,\cdots,m)$. 该模型依权重大小对所有因素均衡兼顾，比较适用于要求总和最大的情形.

4) 模型 IV:$M(\wedge,\oplus)$—— 取小上界和型

运算法则为 $b_j = \min\left\{1, \sum_{i=1}^{n}(a_i \wedge r_{ij})\right\}$ $(j=1,2,\cdots,m)$. 使用该模型时,需要注意的是:各个 a_i 不能取得偏大,否则可能出现 b_j 均等于1的情形;各个 a_i 也不能取得太小,否则可能出现 b_j 均等于各个 a_i 之和的情形,这将使单因素评判的有关信息丢失.

5) 模型 V:$M(\wedge,+)$—— 均衡平均型

运算法则为 $b_j = \sum_{i=1}^{n}\left(a_i \wedge \frac{r_{ij}}{r_0}\right)$ $(j=1,2,\cdots,m)$,其中 $r_0 = \sum_{k=1}^{n} r_{kj}$. 该模型适用于综合评判矩阵 $\boldsymbol{R}$ 中的元素偏大或偏小时的情形.

8.2.2 案例分析

例 8.4 考虑一个服装评判的问题,为此建立因素集 $U=\{u_1,u_2,u_3,u_4\}$,其中 u_1 表示花色,u_2 表示式样,u_3 表示耐穿程度,u_4 表示价格. 建立评判集 $V=\{v_1,v_2,v_3,v_4\}$,其中 v_1 表示很欢迎,v_2 表示较欢迎,v_3 表示不太欢迎,v_4 表示不欢迎. 进行单因素评判的结果如下:

$$u_1 \mapsto r_1 = (0.2,0.5,0.2,0.1), \qquad u_2 \mapsto r_2 = (0.7,0.2,0.1,0)$$

$$u_3 \mapsto r_3 = (0,0.4,0.5,0.1), \qquad u_4 \mapsto r_4 = (0.2,0.3,0.5,0)$$

设有两类顾客,他们根据自己的喜好对各因素所分配的权重分别为

$$\boldsymbol{A}_1 = (0.1,0.2,0.3,0.4), \qquad \boldsymbol{A}_2 = (0.4,0.35,0.15,0.1)$$

试分析这两类顾客对此服装的喜好程度.

分析 由单因素评判构造综合评判矩阵:

$$\boldsymbol{R} = \begin{bmatrix} 0.2 & 0.5 & 0.2 & 0.1 \\ 0.7 & 0.2 & 0.1 & 0 \\ 0 & 0.4 & 0.5 & 0.1 \\ 0.2 & 0.3 & 0.5 & 0 \end{bmatrix}$$

用模型 $M(\wedge,\vee)$ 计算综合评判为

$$\boldsymbol{B}_1 = \boldsymbol{A}_1 \circ \boldsymbol{R} = (0.2,0.3,0.4,0.1), \qquad \boldsymbol{B}_2 = \boldsymbol{A}_2 \circ \boldsymbol{R} = (0.35,0.4,0.2,0.1)$$

根据最大隶属度原则知,第一类顾客对此服装不太欢迎,第二类顾客对此服装则比较欢迎.

由于程序源码过长,读者请登陆华中数学建模网 http://www.shumo.cn/ 大学数学实验栏目下载学习本程序.

程序输出结果如下:

```
ans=
    0.2000    0.3000    0.4000    0.1000
```

```
ans=
     0.3500    0.4000    0.2000    0.1000
```

例 8.5 某校规定,在对一位教师的评价中,若"好"与"较好"占 50%以上,可晋升为教授,教授分教学型教授和科研型教授,在评价指标上给出不同的权重,分别为 $\boldsymbol{A}_1=(0.2,0.5,0.1,0.2)$,$\boldsymbol{A}_2=(0.2,0.1,0.5,0.2)$. 学科评议组由 7 人组成,对该教师的评价见表 8.3,请判别该教师能否晋升,可晋升为哪一级教授.

表 8.3 对该教师的评价

	好	较好	一般	较差	差
政治表现	4	2	1	0	0
教学水平	6	1	0	0	0
科研能力	0	0	5	1	1
外语水平	2	2	1	1	1

分析 将评议组 7 人对每一项的投票按百分比转化成隶属度得综合评判矩阵:

$$\boldsymbol{R}=\begin{bmatrix} 0.57 & 0.29 & 0.14 & 0 & 0 \\ 0.86 & 0.14 & 0 & 0 & 0 \\ 0 & 0 & 0.71 & 0.14 & 0.14 \\ 0.29 & 0.29 & 0.14 & 0.14 & 0.14 \end{bmatrix}$$

按模型 $M(\wedge,\vee)$针对两个权重分别计算得

$$B_1=A_1\circ R=(0.5,0.2,0.14,0.14,0.14)$$

$$B_2=A_2\circ R=(0.2,0.2,0.5,0.14,0.14)$$

由于要计算百分比,需要将上述评判结果进一步归一化如下:

$\boldsymbol{B}'_1=(0.46,0.18,0.12,0.12,0.12)$, $\boldsymbol{B}'_2=(0.17,0.17,0.42,0.12,0.12)$

显然,对第一类权重"好"与"较好"占 50%以上,故该教师可晋升为教学型教授. 程序与例 8.4 相同.

输入及结果:

```
%输入评价指标权重矩阵和综合评判矩阵
A1=[0.2 0.5 0.1 0.2];
A2=[0.2 0.1 0.5 0.2];
R=[0.57 0.29 0.14 0 0;
    0.86 0.140 0 0;
    0 0 0.71 0.14 0.14;
    0.29 0.29 0.14 0.14 0.14];
fuzzy_zhpj(1,A1,R)
fuzzy_zhpj(1,A2,R)
```

程序输出结果如下:

```
ans=
    0.5000    0.2000    0.1400    0.1400    0.1400
ans=
    0.2000    0.2000    0.5000    0.1400    0.1400
```

例 8.6 某产粮区进行耕作制度改革，制定了甲、乙、丙三个方案见表 8.4，以表 8.5 作为评价指标，5 个因素权重定为(0.2,0.1,0.15,0.3,0.25)，请确定应该选择哪一个方案.

表 8.4 三个方案

方案	亩产量/(kg/亩)	产品质量	亩用工量	亩纯收入/元	生态影响
甲	592.5	3	55	72	5
乙	529	2	38	105	3
丙	412	1	32	85	2

表 8.5 5 个评价标准

分数	亩产量	产品质量	亩用工量	亩纯收入	生态影响
5	550～600	1	<20	>130	1
4	500～550	2	20～30	110～130	2
3	450～500	3	30～40	90～110	3
2	400～450	4	40～50	70～90	4
1	350～400	5	50～60	50～70	5
0	<350	6	>60	<50	6

分析 根据评价标准建立各指标的隶属函数如下.

亩产量的隶属函数：

$$C(x_1)=\begin{cases}0, & x_1\leqslant 350\\ \dfrac{x_1-350}{600-350}, & 350<x_1<600\\ 1, & x_1\geqslant 600\end{cases}$$

产品质量的隶属函数：

$$C(x_2)=\begin{cases}1, & x_2\leqslant 1\\ 1-\dfrac{x_2-1}{6-1}, & 1<x_2<6\\ 0, & x_2\geqslant 6\end{cases}$$

亩用工量的隶属函数：

$$C(x_3)=\begin{cases}1, & x_3\leqslant 20\\ 1-\dfrac{x_3-20}{60-20}, & 20<x_3<60\\ 0, & x_3\geqslant 60\end{cases}$$

亩纯收入的隶属函数：

$$C(x_4)=\begin{cases}0, & x_4\leqslant 50\\ \dfrac{x_3-50}{130-50}, & 50<x_4<130\\ 1, & x_4\geqslant 130\end{cases}$$

对生态影响的隶属函数：

$$C(x_5)=\begin{cases}1, & x_5\leqslant 1\\ 1-\dfrac{x_5-1}{6-1}, & 1<x_5<6\\ 0, & x_5\geqslant 6\end{cases}$$

将表 8.4 三个方案中数据代入相应隶属函数算出隶属度，从而得到综合评判矩阵：

$$\boldsymbol{R}=\begin{bmatrix}0.97 & 0.716 & 0.248\\ 0.6 & 0.8 & 1\\ 0.125 & 0.55 & 0.7\\ 0.275 & 0.6875 & 0.4375\\ 0.2 & 0.6 & 0.8\end{bmatrix}$$

根据所给权重按加权平均型计算得

$$\boldsymbol{B}=\boldsymbol{A}\circ\boldsymbol{R}=(0.4053,0.6620,0.5858)$$

根据最大隶属度原则，0.662 最大，所对应的是乙方案，故应选择乙方案.

程序同例 8.4.

输入及结果：

```
%输入评价指标权重矩阵和综合评判矩阵
A=[0.2 0.1 0.15 0.3 0.25];
R=[0.97 0.716 0.248;
   0.6 0.8 1;
   0.125 0.55 0.7;
   0.275 0.6875 0.4375;
   0.2 0.6 0.8];
fuzzy_zhpj(3,A,R)  %调用综合评判函数
```

程序运行结果如下：

```
ans=
    0.4053    0.6620    0.5858
```

例 8.7 表 8.6 是大气污染物评价标准. 今测得某日某地以上污染物日均浓度为(0.07,0.20,0.123,5.00,0.08,0.14),各污染物权重为(0.1,0.20,0.3,0.3,0.05,0.05),试判别其污染等级.

表 8.6 大气污染物评价标准 单位:mg/m^3

污染物	I 级	II 级	III 级	IV 级
SO_2	0.05	0.15	0.25	0.50
TSP	0.12	0.30	0.50	1.00
NO_x	0.10	0.10	0.15	0.30
CO	4.00	4.00	6.00	10.00
PM_1	0.05	0.15	0.25	0.50
O_3	0.12	0.16	0.20	0.40

分析 由于大气中各污染物含量均是越少大气质量越高,可构造各污染物含量对四个等级的隶属函数如下.

对 I 级的隶属函数:

$$r_{i1}=\begin{cases}1, & x_i\leqslant a\\ \dfrac{b-x_i}{b-a}, & a<x_i<b\\ 0, & x_i\geqslant b\end{cases}$$

对 II 级的隶属函数:

$$r_{i2}=\begin{cases}\dfrac{x_i-a}{b-a}, & a<x_i\leqslant b\\ \dfrac{c-x_i}{c-b}, & b<x_i<c\\ 0, & x_i\geqslant c \text{ 或 } x_i\leqslant a\end{cases}$$

对 III 级的隶属函数:

$$r_{i3}=\begin{cases}\dfrac{x_i-b}{c-b}, & b<x_i\leqslant c\\ \dfrac{d-x_i}{d-c}, & c<x_i<d\\ 0, & x_i\geqslant d \text{ 或 } x_i\leqslant b\end{cases}$$

对 IV 级的隶属函数:

$$r_{i4}=\begin{cases}0, & x_i\leqslant c\\ \dfrac{d-x_i}{d-c}, & c<x_i<d\\ 1, & x_i\geqslant d\end{cases}$$

其中 $i=1,2,3,4,5,6$ 表示 6 种污染物,如 r_{24} 表示第二种污染物的含量 x_i 对 IV 级的

隶属度，而 a,b,c,d 依次表示评价标准中各污染物含量.

对污染物 SO_2，其含量 $x_i=0.07$，计算其对各等级的隶属度如下：因 $0.05<0.07<0.15$，故

$$r_{11}=\frac{0.15-0.07}{0.15-0.05}=0.8, \qquad r_{12}=\frac{0.07-0.05}{0.15-0.05}=0.2$$

因 $0.07<0.15$，故 $r_{13}=0$，因 $0.07<0.25$，故 $r_{14}=0$.

同理可计算其他污染物含量对各等级的隶属度，从而得综合评判矩阵：

$$\boldsymbol{R}=\begin{bmatrix} 0.8 & 0.2 & 0 & 0 \\ 0.56 & 0.44 & 0 & 0 \\ 0 & 0.6 & 0.4 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0.7 & 0.3 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \end{bmatrix}$$

结合权重，选择加权平均型进行计算得 $\boldsymbol{B}=\boldsymbol{A}\circ\boldsymbol{R}=(0.252,0.478,0.27,0)$，根据最大隶属度原则，0.478 最大，故当日大气质量为 II 级.

程序同例 8.4.

输入及结果：

```
A=[0.1 0.2 0.3 0.3 0.05 0.05];
R=[0.8 0.2 0 0;
   0.56 0.44 0 0;
   0 0.6 0.4 0;
   0 0.5 0.5 0;
   0.7 0.3 0 0;
   0.5 0.5 0 0];
fuzzy_zhpj(3,A,R)
```

程序运行结果如下：

```
ans=
    0.2520    0.4780    0.2700         0
```

8.2.3 方法评论

模糊综合评判经常用来处理一类选择和排序的问题. 应用的关键在于模糊综合评判矩阵的建立，它是由单因素评判向量所构成的，简单的情形可按类似于百分比的方式得到，稍复杂一点的情形需要构造隶属函数来进行转化，这一点类似于前面的模糊模式识别，要注意评判指标的属性，合理选择隶属函数. 进行综合评判时，要根据问题的实际情况，选择恰当的模型来进行计算. 另外，关于权重，前面都是直接给出来的，而在实际当中是不会有的. 当然，读者可以自行设定，但如若能用到一些数学方

法，如层次分析法等，将定性和定量结合，则会显得更有说服力.

8.3 模糊聚类分析

在科学技术、经济管理中常常需要按一定的标准进行分类. 例如，根据大气中的成分对大气污染状态等级分类，根据各门课程的学习成绩对学生进行划分等. 对所研究事物按一定标准进行分类的方法称为聚类分析. 由于科学技术、经济管理中的分类界限往往不分明，因此，在实际中常采用模糊聚类分析方法.

8.3.1 理论介绍

1. 相关定义定理介绍

定义 8.1 设 $\boldsymbol{R}=(r_{ij})_{m\times n}$ $(0\leqslant r_{ij}\leqslant 1)$，则称 $\boldsymbol{R}$ 为模糊矩阵. 当 r_{ij} 只取 0 或 1 时，称 $\boldsymbol{R}$ 为布尔矩阵.

定义 8.2 设 $\boldsymbol{R}=(r_{ij})_{m\times n}$ 为模糊矩阵，$\forall\lambda\in[0,1]$，称 $\boldsymbol{R}_\lambda=(r_{ij}^{(\lambda)})_{m\times n}$ 为 $\boldsymbol{R}$ 的 λ-截矩阵. 其中，$r_{ij}^{(\lambda)}=\begin{cases}1, & r_{ij}\geqslant a,\\ 0, & r_{ij}<\lambda.\end{cases}$

定义 8.3 设 $\mathbf{R}=(r_{ij})_{n\times n}$ 是 n 阶模糊方阵，$\boldsymbol{I}$ 是 n 阶单位方阵，若 $\boldsymbol{R}$ 满足

(1) 自反性：$\boldsymbol{I}\leqslant\boldsymbol{R}(\Leftrightarrow r_{ij}=1)$；

(2) 对称性：$\boldsymbol{R}^{\mathrm{T}}=\boldsymbol{R}(\Leftrightarrow r_{ij}=r_{ji})$；

(3) 传递性：$\boldsymbol{R}^2\leqslant\boldsymbol{R}(\Leftrightarrow\max\{(r_{ik}\wedge r_{kj})\mid 1\leqslant k\leqslant n\}\leqslant r_{ij})$.

则称 $\boldsymbol{R}$ 为模糊等价矩阵.

定义 8.4 设 $\boldsymbol{R}=(r_{ij})_{m\times n}$ 是 n 阶模糊方阵，$\boldsymbol{I}$ 是 n 阶单位方阵，若 $\boldsymbol{R}$ 满足

(1) 自反性：$\boldsymbol{I}\leqslant\boldsymbol{R}(\Leftrightarrow r_{ij}=1)$；

(2) 对称性：$\boldsymbol{R}^{\mathrm{T}}=\boldsymbol{R}(\Leftrightarrow r_{ij}=r_{ji})$；.

则称 $\boldsymbol{R}$ 为模糊相似矩阵.

定义 8.5 设 $\boldsymbol{A}=(a_{ij})_{m\times s}$，$\boldsymbol{B}=(b_{ij})_{s\times n}$，称 $(c_{ij})_{m\times n}=\boldsymbol{A}\circ\boldsymbol{B}$ 为 $\boldsymbol{A}$ 与 $\boldsymbol{B}$ 的合成，其中，$c_{ij}=\max\{(a_{ik}\wedge b_{kj})\mid 1\leqslant k\leqslant s\}$.

定理 8.1 设 $\boldsymbol{R}$ 是 n 阶模糊等价矩阵，则 $\forall\,0\leqslant\lambda<\mu\leqslant 1$，$\boldsymbol{R}_\mu$ 所决定的分类中的每一个类是 $\boldsymbol{R}_\lambda$ 所决定的分类中的某个子类.

该定理表明，当 $\lambda<\mu$ 时，$\boldsymbol{R}_\mu$ 的分类是 $\boldsymbol{R}_\lambda$ 的分类的加细，当 λ 由 1 变到 0 时，$\boldsymbol{R}_\lambda$ 的分类由细变粗，形成一个动态的聚类图. 但该定理针对的是模糊等价矩阵，而实际中得到的往往是一个模糊相似矩阵，对于模糊相似矩阵，通常采用的是求传递闭包的方法.

定理 8.2 设 $\boldsymbol{R}$ 是 n 阶模糊相似矩阵，则存在一个最小的自然数 $k(k\leqslant n)$，使得 $\boldsymbol{R}^k$ 为模糊等价矩阵，且对一切大于 k 的自然数 l，恒有 $\boldsymbol{R}^l=\boldsymbol{R}^k$. $\boldsymbol{R}^k$ 称为 $\boldsymbol{R}$ 的传递闭包

矩阵.

传递闭包矩阵通常记为 $t(\boldsymbol{R})$，一个实用的简捷方法为二次方法，即从模糊相似矩阵 $\boldsymbol{R}$ 出发，依次求二次方，即

$$\boldsymbol{R}\to\boldsymbol{R}^2\to\boldsymbol{R}^4\to\cdots\to\boldsymbol{R}^{2^i}\to\cdots$$

直到第一次出现 $\boldsymbol{R}^k\circ\boldsymbol{R}^k=\boldsymbol{R}^k$ 时，即有 $t(\boldsymbol{R})=\boldsymbol{R}^k$.

2. 聚类步骤

进行模糊聚类分析，通常分为以下几个步骤：

(1) 建立数据矩阵.

设论域 $U=\{u_1,u_2,\cdots,u_n\}$ 为被分类对象，每个对象又有 m 个指标表示其性状：

$$\boldsymbol{x}_i=\{x_{i1},x_{i2},\cdots,x_{im}\},\quad i=1,2,\cdots,n$$

则得到原始数据矩阵 $\boldsymbol{X}=(x_{ij})_{n\times m}$.

在实际应用中，不同的数据有不同的量纲，为了使有不同量纲的量能进行比较，需要将数据规格化，以下是常用的数据规格化方法.

① 标准差标准化：

$$x'_{ij}=\frac{x_{ij}-\overline{x_j}}{s_j},\quad i=1,2,\cdots,n;j=1,2,\cdots,m$$

其中$\overline{x_j}=\dfrac{1}{n}\sum\limits_{i=1}^{n}x_{ij}$，$s_j=\sqrt{\dfrac{1}{n}\sum\limits_{i=1}^{n}(x_{ij}-\overline{x_j})^2}$.

② 极差正规化：

$$x'_{ij}=\frac{x_{ij}-\min\limits_{1\leqslant k\leqslant n}\{x_{kj}\}}{\max\limits_{1\leqslant k\leqslant n}\{x_{kj}\}-\min\limits_{1\leqslant k\leqslant n}\{k_{kj}\}},\quad i=1,2,\cdots,n;j=1,2,\cdots,m$$

③ 极差标准化：

$$x'_{ij}=\frac{x_{ij}-\overline{x_j}}{\max\limits_{1\leqslant k\leqslant n}\{x_{kj}\}-\min\limits_{1\leqslant k\leqslant n}\{x_{kj}\}},\quad i=1,2,\cdots,n;j=1,2,\cdots,m$$

④ 最大值规格化：

$$x'_{ij}=\frac{x_{ij}}{\max\limits_{1\leqslant k\leqslant n}\{x_{kj}\}},\quad i=1,2,\cdots,n;j=1,2,\cdots,m$$

(2) 建立模糊相似矩阵.

建立 x_i 与 x_j 相似程度 r_{ij} 通常有以下几种方法.

① 相似系数法. 相似系数法可分为夹角余弦法和相关系数法.

(a) 夹角余弦法：

$$r_{ij}=\frac{\sum\limits_{k=1}^{m}x_{ik}x_{jk}}{\sqrt{\sum\limits_{k=1}^{m}x_{ik}^2}\sqrt{\sum\limits_{k=1}^{m}x_{jk}^2}}$$

(b) 相关系数法：
$$r_{ij}=\frac{\sum_{k=1}^{m}\left|x_{ik}-\overline{x_i}\right|\left|x_{jk}-\overline{x_j}\right|}{\sqrt{\sum_{k=1}^{m}x_{ik}^2}\sqrt{\sum_{k=1}^{m}x_{jk}^2}}$$

② 距离法. 一般地，采用距离法时 $r_{ij}=1-c(d(x_i,x_j))^\alpha$，其中 c,α 为适当选取的参数，使得 $0\leqslant r_{ij}\leqslant 1$ 成立. 距离法中距离 $d(x_i,x_j)$ 的计算包括以下几种方法.

(a) 欧几里得距离： $d(x_i,x_j)=\sqrt{\sum_{k=1}^{m}(x_{ik}-x_{jk})^2}$

(b) 海明距离： $d(x_i,x_j)=\sum_{k=1}^{m}\left|x_{ik}-x_{jk}\right|$

(c) 切比雪夫距离： $d(x_i,x_j)=\max_{1\leqslant k\leqslant m}\left|x_{ik}-x_{jk}\right|$

③ 贴近度法. 常用的贴近度法包括最大最小法、算术平均最小法和几何平均最小法，具体如下.

(a) 最大最小法：
$$r_{ij}=\frac{\sum_{k=1}^{m}(x_{ik}\wedge x_{jk})}{\sum_{k=1}^{m}(x_{ik}\vee x_{jk})}$$

(b) 算术平均最小法：
$$r_{ij}=\frac{\sum_{k=1}^{m}(x_{ik}\wedge x_{jk})}{\frac{1}{2}\sum_{k=1}^{m}(x_{ik}+x_{jk})}$$

(c) 几何平均最小法：
$$r_{ij}=\frac{\sum_{k=1}^{m}(x_{ik}\wedge x_{jk})}{\sum_{k=1}^{m}\sqrt{x_{ik}x_{jk}}}$$

(3) 聚类并画出动态聚类图.

进行模糊聚类的方法有许多种，但一般常用的是传递闭包法，其聚类步骤通常为

① 求出模糊相似矩阵 $\boldsymbol{R}$ 的传递闭包矩阵 $t(\boldsymbol{R})$；

② 按 λ 由大到小进行聚类；

③ 画出动态聚类图.

2. 案例分析

例 8.8 设 $U=\{x_1,x_2,x_3,x_4,x_5\}$，现有与其对应的模糊等价矩阵如下，试根据模糊等价矩阵对 U 中元素进行分类.

$$R=\begin{bmatrix}1 & 0.4 & 0.8 & 0.5 & 0.5\\ 0.4 & 1 & 0.4 & 0.4 & 0.4\\ 0.8 & 0.4 & 1 & 0.5 & 0.5\\ 0.5 & 0.4 & 0.5 & 1 & 0.6\\ 0.5 & 0.4 & 0.5 & 0.6 & 1\end{bmatrix}$$

分析 依次取$\lambda=1.0.8,0.6,0.5,0.4$，得$\lambda$-截矩阵及分类结果如下：

$$R_1=\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

U 分为 5 类：$\{x_1\},\{x_2\},\{x_3\},\{x_4\},\{x_5\}$；

$$R_{0.8}=\begin{bmatrix}1 & 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 0\\ 1 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

U 分为 4 类：$\{x_1,x_3\},\{x_2\},\{x_4\},\{x_5\}$；

$$R_{0.6}=\begin{bmatrix}1 & 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 0\\ 1 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 1\\ 0 & 0 & 0 & 1 & 1\end{bmatrix}$$

U 分为 3 类：$\{x_1,x_3\},\{x_2\},\{x_4,x_5\}$；

$$R_{0.5}=\begin{bmatrix}1 & 0 & 1 & 1 & 1\\ 0 & 1 & 0 & 0 & 0\\ 1 & 0 & 1 & 1 & 1\\ 1 & 0 & 1 & 1 & 1\\ 1 & 0 & 1 & 1 & 1\end{bmatrix}$$

U 分为 2 类：$X=\{x_1,x_3,x_4,x_5\},\{x_2\}$；

$$R_{0.4}=\begin{bmatrix}1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\end{bmatrix}$$

U 分为 1 类：$X=\{x_1,x_2,x_3,x_4,x_5\}$.

由于程序源码过长，读者请登陆华中数学建模网 http://www.shumo.cn/大学数学实验栏目下载学习本程序.

由于程序运行结果太长，这里仅输出部分如下：

```
lamd=

    0.6000
```

所对应的截矩阵为

```
Alamd=

    1    0    1    0    0
    0    1    0    0    0
    1    0    1    0    0
    0    0    0    1    1
    0    0    0    1    1
C=

    1    3
    2    0
    4    5
```

上述结果中 Alamd 为 $\lambda=0.6$ 时的入一截矩阵，$\boldsymbol{C}$ 分三行，表示分类结果为三类，第一行的 1,3 表示 x_1,x_3 为一类，同理，第二行说明 x_2 为一类，第三行则说明 x_4,x_5 为一类.

例 8.9 设有模糊相似矩阵如下，试求其传递闭包.

$$\boldsymbol{R}=\begin{bmatrix}0 & 0.1 & 0.2\\ 0.1 & 1 & 0.3\\ 0.2 & 0.3 & 1\end{bmatrix}$$

分析

$$\boldsymbol{R}^2=\boldsymbol{R}\circ\boldsymbol{R}=\begin{bmatrix}1 & 0.2 & 0.2\\ 0.2 & 1 & 0.3\\ 0.2 & 0.3 & 0.2\end{bmatrix},\qquad \boldsymbol{R}^4=\boldsymbol{R}^2\circ\boldsymbol{R}^2=\begin{bmatrix}1 & 0.2 & 0.2\\ 0.2 & 1 & 0.3\\ 0.2 & 0.3 & 0.2\end{bmatrix}$$

由于 $\boldsymbol{R}^2=\boldsymbol{R}^4$，故传递闭包矩阵 $t(\boldsymbol{R})=\boldsymbol{R}^2$.

求传递闭包的 M-函数：

```
function[A]=fuzzy_cdbb(R)  %由模糊相似矩阵求传递闭包
js0=0;
while(1)
A=Max_Min(R,R);
js0=js0+ 1;
if(A==R)
    break;
```

```
    else
        R=A;
    end
    end
```

输入及结果：

```
% 输入原始数据,调用函数
R=[1 0.1 0.2;0.1 1 0.3;0.2 0.3 1];
[A]=fuzzy_cdbb(R)
```

运行结果如下：

```
A=
    1.0000    0.2000    0.2000
    0.2000    1.0000    0.3000
    0.2000    0.3000    1.0000
```

例 8.10 某环保部门拟对该地区 5 个环境区域 $X=\{x_1,x_2,x_3,x_4,x_5\}$按污染情况进行分类.设每个区域包括空气、水分、土壤和作物 4 个因素,现测得 5 个环境区域的污染数据为 $x_1=(80,10,6,2)$,$x_2=(50,1,6,4)$,$x_3=(90,6,4,6)$,$x_4=(40,5,7,3)$,$x_5=(10,1,2,4)$,试对 X 进行分类.

分析 由题设构造特性指标矩阵为

$$\boldsymbol{X}^*=\begin{bmatrix}80 & 10 & 6 & 2\\ 50 & 1 & 6 & 4\\ 90 & 6 & 4 & 6\\ 40 & 5 & 7 & 3\\ 10 & 1 & 2 & 4\end{bmatrix}$$

采用最大值规格化法将原始数据规格化为

$$\boldsymbol{X}=\begin{bmatrix}0.89 & 1 & 0.86 & 0.33\\ 0.56 & 0.1 & 0.86 & 0.67\\ 1 & 0.6 & 0.57 & 1\\ 0.44 & 0.5 & 1 & 0.5\\ 0.11 & 0.1 & 0.29 & 0.67\end{bmatrix}$$

用最大最小法构造模糊相似矩阵得

$$\boldsymbol{R}=\begin{bmatrix}1 & 0.54 & 0.62 & 0.63 & 0.24\\ 0.54 & 1 & 0.55 & 0.70 & 0.53\\ 0.62 & 0.55 & 1 & 0.56 & 0.37\\ 0.63 & 0.70 & 0.56 & 1 & 0.38\\ 0.24 & 0.53 & 0.37 & 0.38 & 1\end{bmatrix}$$

用平方法合成传递闭包得

$$t(\boldsymbol{R})=\boldsymbol{R}^4=\begin{bmatrix}1 & 0.63 & 0.62 & 0.53 & 0.53\\ 0.63 & 1 & 0.62 & 0.70 & 0.53\\ 0.62 & 0.62 & 1 & 0.62 & 0.53\\ 0.63 & 0.70 & 0.62 & 1 & 0.53\\ 0.53 & 0.53 & 0.53 & 0.53 & 1\end{bmatrix}$$

将 $t(\boldsymbol{R})$中的元素从大到小排列为 $1>0.70>0.63>0.62>0.53$.

取 $\lambda=1$,得

$$t(\boldsymbol{R})_1=\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

X 被分成 5 类:$\{x_1\},\{x_2\},\{x_3\},\{x_4\},\{x_5\}$;

取 $\lambda=0.70$,得

$$t(\boldsymbol{R})_{0.70}=\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 1 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

X 被分成 4 类:$\{x_1\},\{x_2,x_4\},\{x_3\},\{x_5\}$;

取 $\lambda=0.63$,得

$$t(\boldsymbol{R})_{0.63}=\begin{bmatrix}1 & 1 & 0 & 1 & 0\\ 1 & 1 & 0 & 1 & 0\\ 0 & 0 & 1 & 0 & 0\\ 1 & 1 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

X 被分成 3 类:$\{x_1,x_2,x_4\},\{x_3\},\{x_5\}$;

取 $\lambda=0.62$,得

$$t(\boldsymbol{R})_{0.62}=\begin{bmatrix}1 & 1 & 1 & 1 & 0\\ 1 & 1 & 1 & 1 & 0\\ 1 & 1 & 1 & 1 & 0\\ 1 & 1 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

X 被分成 2 类:$\{x_1,x_2,x_3,x_4\},\{x_5\}$;

取 $\lambda=0.53$,得

$$t(\boldsymbol{R})_{0.53}=\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix}$$

X 被分成 1 类：$\{x_1,x_2,x_3,x_4,x_5\}$.

上述过程用动态聚类图表示如图 8.6 所示.

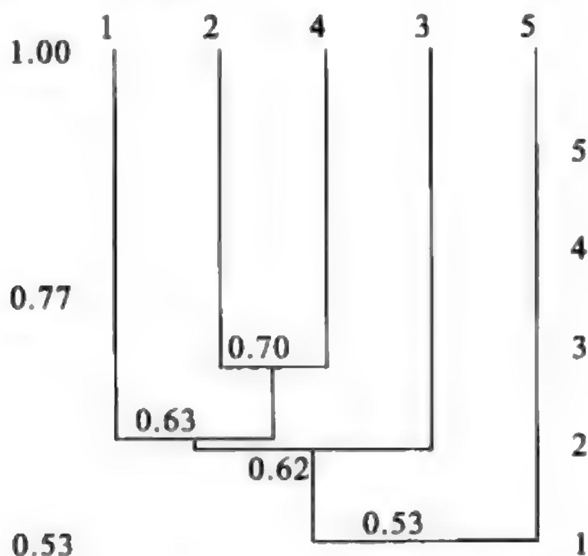


图 8.1　动态聚类图

主程序见华中数学建模网 http://www.shumo.cn/大学数学实验栏目.

(1) 将数据标准化的 M-函数：

```
function[X]=F_JlSjBzh(cs,X)   %定义函数
%模糊聚类分析数据标准化变换:[X]=F_JlSjBzh(cs,X)
%X,数据矩阵
%cs=0,不变换;cs=1,标准差变换;cs=2,极差变换;cs=其他,最大值规格化
```

(2) 求相似矩阵的 M-函数：

```
function[R]=F_JlR(cs,X)   %定义函数
%模糊聚类分析建立模糊相似矩阵:[R]=F_JlR(cs,X)
%X,数据矩阵
%cs=1,数量积法
%cs=2,夹角余弦法
%cs=3,相关系数法
%cs=4,指数相似系数法
%cs=5,最大最小法
%cs=6,算术平均最小法
%cs=7,几何平均最小法
%cs=8,一般欧式距离法
%cs=9,一般海明距离法
```

```
%cs=10,一般切比雪夫距离法
%cs=11,倒数欧式距离法
%cs=12,倒数海明距离法
%cs=13,倒数切比雪夫距离法
%cs=14,指数欧式距离法
%cs=15,指数海明距离法
%cs=16,指数切比雪夫距离法
```

下面这段程序是表示求模糊矩阵的乘法：

```
function[C]=Max_Min(A,B)%
[m,s]=size(A);[s1,n]=size(B);C=[];
if(s1~=s)return;end
for(i =1:m)for(j=1:n)C(i,j)=0;
      for(k =1:s)x=0;
            if(A(i,k)<B(k,j))x=A(i,k);
            else x=B(k,j);end
            if(C(i,j)<x)C(i,j)=x;end
   end
end;end
```

下面是画动态聚类图：

```
function[M,N]=F_JlDtjl(R)
%N 为元胞数组,保存聚类的结果
%R 为相似矩阵
```

(3) 聚类分析的主函数：

```
function fuzzy_jlfx(bzh,fa,X)  %得到聚类结果
[X]=F_JlSjBzh(bzh,X);  %数据标准化
[R]=F_JlR(fa,X);  %建立相似矩阵
[M,N]=F_JlDtjl(R)  %动态聚类并画出聚类图
```

输入及结果：

```
%输入原始数据,调用主函数
X=[80 10 6 2;50 1 6 4;90 6 4 6;40 5 7 3;10 1 2 4];
fuzzy_jlfx(3,5,X)
```

运行结果见前面解题步骤.

8.3.2 方法评论

这里进行模糊聚类所采取的是传递闭包的方法，其特点是步骤清晰，易于编程实现. 当然，对于一些规模比较小的模糊聚类问题，也可采用其他方法如直接聚类法、最大树法及编网法等来实现，这些方法在一些模糊数学教材中都有所提及. 读者在运用

上述程序时,需要将所有的函数存放在同一目录下,因为他们之间存在一些函数间的互相调用,这样,在路径正确的前提下才能正常运行.另外,在一些教材中提到了最佳分类方案,而在实际应用中,所谓的最佳分类方案很多时候是不可取的,因此,不必强求于所谓的最佳分类,而应着眼于问题的实际,需要分为几类,分为几类比较合理来进行诀择.

8.4 模糊线性规划

线性规划是最优化方法中理论完整、方法成熟、应用广泛的一个重要分支,可以用于生产计划、物资调运、资源优化配置、地区经济规划等问题.普通线性规划约束条件和目标函数都是确定的,但在一些实际问题中,约束条件可能带有弹性,目标函数可能不是单一的,价值系数可能带有模糊性.这些问题可以用模糊数学的方法来处理.引用隶属函数概念,将约束条件和目标函数模糊化,从而导出一个新的线性规划问题——模糊线性规划.

8.4.1 理论介绍

1. 线性规划与模糊线性规划

普通线性规划的标准形式为

$$\min f=t_0(x)$$
$$\text{s.t.}\begin{cases}t_i(x)=b_i, & i=1,2,\cdots,m\\ x\geqslant 0\end{cases}\tag{8.4.1}$$

其中,$\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$,$t_0(\boldsymbol{x})=c_1x_1+c_2x_2+\cdots+c_nx_n$,

$$t_i(\boldsymbol{x})=a_{i1}x_1+a_{i2}x_2+\cdots+a_{in}x_n,\quad i=1,2,\cdots,m$$

约束条件带有弹性的模糊线性规划为

$$\min f=t_0(\boldsymbol{x})$$
$$\text{s.t.}\begin{cases}t_i(\boldsymbol{x})=[b_i,d_i], & i=1,2,\cdots,m\\ \boldsymbol{x}\geqslant \boldsymbol{0}\end{cases}\tag{8.4.2}$$

其中,$t_i(\boldsymbol{x})=[b_i,d_i]$表示当 $d_i=0$ 时,$t_i(\boldsymbol{x})=b_i$.当 $d_i>0$ 时 $t_i(\boldsymbol{x})$取$[b_i-d_i,b_i+d_i]$内某一值.d_i称为伸缩指标,其他符号说明同(8.4.1)式.

另一个与模糊线性规划(8.4.2)有关的普通线性规划为

$$\min f=t_0(\boldsymbol{x})$$
$$\text{s.t.}\begin{cases}b_i-d_i\leqslant t_i(\boldsymbol{x})\leqslant b_i+d_i, & i=1,2,\cdots,m\\ \boldsymbol{x}\geqslant \boldsymbol{0}\end{cases}\tag{8.4.3}$$

与(8.4.2)式相比,(8.4.3)式中约束条件只要求 $t_i(\boldsymbol{x})$在$[b_i-d_i,b_i+d_i]$内取值即可,但(8.4.2)式中约束条件表明 $t_i(\boldsymbol{x})$不仅要在$[b_i-d_i,b_i+d_i]$内取值,并且尽可

能不要偏离 b_i 太远. 因此(8.4.2)式中约束条件比(8.4.3)式中更强，而(8.4.1)式中约束条件比(8.4.2)式中更强. 就目标函数的最优值而言，则是(8.4.1)式中最大，(8.4.2)式中其次，(8.4.3)式中最小.

2. 模糊线性规划的求解

解模糊线性规划的基本思想是化模糊线性规划为普通线性规划，因此，关键是如何把带有伸缩指标的约束条件转化为普通约束条件. 方法是引入隶属函数.

定义模糊线性规划(8.4.2)中带有弹性的约束条件的隶属函数为

$$A_i(\boldsymbol{x})=1-\frac{|t_i(\boldsymbol{x})-b_i|}{d_i}, \quad b_i-d_i\leqslant t_i(\boldsymbol{x})\leqslant b_i+d_i$$

当 $t_i(\boldsymbol{x})=b_i$ 时，$A_i(\boldsymbol{x})=1$，即为(8.4.1)式中普通约束条件. 为了表达 $t_i(\boldsymbol{x})$ 尽可能不要偏离 b_i 太远，因此，希望 $A_i(\boldsymbol{x})$ 尽可能的大. 但到底大到多少，这个要结合所有约束条件和目标函数来决非定，为此，引入一个 $\lambda\in[0,1]$，让 $A_i(\boldsymbol{x})\geqslant\lambda$，如果 λ 能够充分大，则 $A_i(\boldsymbol{x})$ 就能尽可能的大. 此时，结合隶属函数的定义有

$$d_i\lambda-d_i\leqslant t_i(x)-b_i\leqslant d_i-d_i\lambda$$

若普通线性规划(8.4.1)和(8.4.3)的最优值分别为 f_0，f_1，则模糊线性规划(8.4.2)取值情况最好是 f_1，最差是 f_0，因此，可认为其伸缩指标为 $d_0=f_0-f_1$，并且希望其最优值尽可能偏离 f_0 远些或说尽可能离 f_1 近些. 当然，在一些实际问题中也可人为决定目标函数的伸缩指标. 为此，定义模糊线性规划(8.4.2)中目标函数的隶属函数为

$$G(\boldsymbol{x})=\frac{f_0-t_0(\boldsymbol{x})}{d_0}, \quad f_0-d_0\leqslant t_0(\boldsymbol{x})\leqslant f_0$$

类似于约束条件，这里也希望 $G(\boldsymbol{x})$ 越大越好，为此，同样引入一个 $\lambda\in[0,1]$，让 $G(\boldsymbol{x})\geqslant\lambda$，则当 λ 充分大时，也能保证 $t_0(\boldsymbol{x})$ 充分接近 f_1. 此时，结合隶属函数的定义有

$$t_0(\boldsymbol{x})+d_0\lambda\leqslant f_0$$

要求模糊线性规划(8.4.2)的最优解 $\boldsymbol{x}^*$，则要求使所有约束条件和目标函数的隶属函数值尽可能达到最大，即求 $\boldsymbol{x}^*$ 满足 $A_i(\boldsymbol{x})\geqslant\lambda$ 及 $G(\boldsymbol{x})\geqslant\lambda$，且使 λ 达到最大值，即解普通线性规划：

$$\max \lambda$$
$$\text{s.t.}\begin{cases}t_0(\boldsymbol{x})+d_0\lambda\leqslant f_0\\ d_i\lambda-d_i\leqslant t_i(\boldsymbol{x})-b_i\leqslant d_i-d_i\lambda, \quad i=1,2,\cdots,m\\ \boldsymbol{x}\geqslant\boldsymbol{0},\lambda\geqslant 0\end{cases} \tag{8.4.4}$$

通过上面的分析知道，模糊线性规划(8.4.2)的求解是通过三个普通线性规划(8.4.1)、(8.4.3)、(8.4.4)的依次求解得到的.

8.4.2 案例分析

例 8.11 求解下面模糊线性规划：

$$\max f=x_1-4x_2+6x_3$$

$$\text{s.t.}\begin{cases}x_1+x_2+x_3\leqslant[8,2]\\x_1-6x_2+x_3\geqslant[6,1]\\x_1-3x_2-x_3=[-4,0.5]\\x_1,x_2,x_3\geqslant 0\end{cases}$$

分析 首先求解普通线性规划：

$$\max f=x_1-4x_2+6x_3$$

$$\text{s.t.}\begin{cases}x_1+x_2+x_3\leqslant 8\\x_1-6x_2+x_3\geqslant 6\\x_1-3x_2-x_3=-4\\x_1,x_2,x_3\geqslant 0\end{cases}$$

得最优解：$x_1=2,x_2=0,x_3=6$，最优值：38；其次，解有如下伸缩指标的普通线性规划：

$$\max f=x_1-4x_2+6x_3$$

$$\text{s.t.}\begin{cases}x_1+x_2+x_3\leqslant 8+2=10\\x_1-6x_2+x_3\geqslant 6-1=5\\x_1-3x_2-x_3\leqslant -4+0.5=-3.5\\x_1-3x_2-x_3\geqslant -4-0.5=-4.5\\x_1,x_2,x_3\geqslant 0\end{cases}$$

得最优解：$x_1=2.75,x_2=0,x_3=7.25$，最优值：46.25；最后，添加新的变量 λ，求解普通线性规划：

$$\max \lambda$$

$$\text{s.t.}\begin{cases}x_1-4x_2+6x_3-(46.25-38)\lambda\geqslant 38\\x_1+x_2+x_3+2\lambda\leqslant 10\\x_1-6x_2+x_3-\lambda\geqslant 5\\x_1-3x_2-x_3+0.5\lambda\leqslant -3.5\\x_1-3x_2-x_3-0.5\lambda\geqslant -4.5\\x_1,x_2,x_3,\lambda\geqslant 0\end{cases}$$

用 LINGO 求解得最优解：$x_1=2.375,x_2=0,x_3=6.625$，最优值为 42.125.

例 8.12 某种饮料含有 3 种主要成分 A_1,A_2,A_2，每瓶含量分别为 75±5 mg，120±5 mg，138±10 mg，这三种成分主要来自于 5 种原料 B_1,B_2,B_3,B_4,B_5. 各种原

料每千克所含的成分与单价见表 8.7. 若生产此种饮料 10000 瓶，如何选择原料成本最小？

表 8.7　原料成分含量与单价

原料	B_1	B_2	B_3	B_4	B_5
A_1/mg	85	60	120	80	120
A_2/mg	80	150	90	160	60
A_3/mg	100	120	150	120	200
单价/元	1.3	1.5	1.6	1.7	1.8

分析　设一瓶饮料需选用 5 种原料分别为 x_1,x_2,x_3,x_4,x_5，则问题可转化为如下模糊线性规划问题：

$$\min f=1.3x_1+1.5x_2+1.6x_3+1.7x_4+1.8x_5$$

$$\text{s.t.}\begin{cases}85x_1+60x_2+120x_3+80x_4+120x_5=[75,5]\\80x_1+150x_2+90x_3+160x_4+60x_5=[120,5]\\100x_1+120x_2+150x_3+120x_4+200x_5=[138,10]\\x_1,x_2,x_3,x_4,x_5\geqslant 0\end{cases}$$

首先求解普通线性规划：

$$\min f=1.3x_1+1.5x_2+1.6x_3+1.7x_4+1.8x_5$$

$$\text{s.t.}\begin{cases}85x_1+60x_2+120x_3+80x_4+120x_5=75\\80x_1+150x_2+90x_3+160x_4+60x_5=120\\100x_1+120x_2+150x_3+120x_4+200x_5=138\\x_1,x_2,x_3,x_4,x_5\geqslant 0\end{cases}$$

得最优解：$x_1=0,x_2=0.6841,x_3=0.0136,x_4=0,x_5=0.2693$，最优值：1.5327；其次，解如下有伸缩指标的普通线性规划：

$$\min f=1.3x_1+1.5x_2+1.6x_3+1.7x_4+1.8x_5$$

$$\text{s.t.}\begin{cases}85x_1+60x_2+120x_3+80x_4+120x_5\leqslant 80\\85x_1+60x_2+120x_3+80x_4+120x_5\geqslant 70\\80x_1+150x_2+90x_3+160x_4+60x_5\leqslant 125\\80x_1+150x_2+90x_3+160x_4+60x_5\geqslant 115\\100x_1+120x_2+150x_3+120x_4+200x_5\leqslant 148\\100x_1+120x_2+150x_3+120x_4+200x_5\geqslant 128\\x_1,x_2,x_3,x_4,x_5\geqslant 0\end{cases}$$

得最优解：$x_1=0,x_2=0.6576,x_3=0.0364,x_4=0,x_5=0.2181$，最优值：1.4373；最后，添加新的变量 λ，求解普通线性规划：

$$
\max \lambda
$$

$$
\text{s.t.}\begin{cases}1.3x_1+1.5x_2+1.6x_3+1.7x_4+1.8x_5+(1.5327-1.4373)\lambda\leqslant 1.5327\\85x_1+60x_2+120x_3+80x_4+120x_5+5\lambda\leqslant 80\\85x_1+60x_2+120x_3+80x_4+120x_5-5\lambda\geqslant 70\\80x_1+150x_2+90x_3+160x_4+60x_5+5\lambda\leqslant 125\\80x_1+150x_2+90x_3+160x_4+60x_5-5\lambda\geqslant 115\\100x_1+120x_2+150x_3+120x_4+200x_5+10\lambda\leqslant 148\\100x_1+120x_2+150x_3+120x_4+200x_5-10\lambda\geqslant 128\\x_1,x_2,x_3,x_4,x_5,\lambda\geqslant 0\end{cases}
$$

用 LINGO 求解得最优解：$x_1=0$，$x_2=0.6708$，$x_3=0.025$，$x_4=0$，$x_5=0.2438$，最优值为 1.485. 由于生产 10000 瓶，故生产总成本为 14850 元.

8.4.3 方法评论

模糊线性规划比一般的线性规划在实际当中应用更强一些，由于不确定性导致约束中的等式和不等式右边的数值可能会出现一定的伸缩，或允许左边的表达式有这样一个伸缩，像这样一类问题，如果仍坚持用普通线性规划建模和求解，显然不太恰当. 关于模糊线性规划的求解，关键是掌握如何依次将其转化为三个普通线性规划.

第 9 章　其他建模方法

9.1　神经网络

9.1.1　人工神经网络

人工神经网络是一门比较年轻的学科，在信号处理、数据挖掘、智能控制等领域中取得了越来越多的成功，已经成为对人类智能模拟的一种重要的方法，有着广泛的应用前景. 近些年来，在中国和美国开展的大学生数学建模竞赛中，也有不少参赛选手使用人工神经网络方法解题，引起广泛的关注. 本节将介绍人工神经网络的一些基本概念和方法，并举例说明它在数学建模中的应用.

人工神经网络是一种用大量处理单元广泛连接组成的人工网络，来模拟人的大脑神经系统.

神经网络的基础在于神经元. 神经元是以生物神经系统的神经细胞为基础的生物模型. 在人们对生物神经系统进行研究，以探讨人工智能的机制时，把神经元数学化，从而产生了神经元数学模型. 大量形式相同的神经元联结在一起就组成了神经网络. 神经网络是一个高度非线性动力学系统. 虽然每个神经元的结构和功能都不复杂，但是整个神经网络的动态行为却是十分复杂的. 因此，用神经网络可以表达实际物理世界的各种现象.

神经网络模型是以神经元的数学模型为基础来描述的. 神经网络模型由网络结构、结点特点和学习规则来表示. 神经网络对人们的巨大吸引力主要有下列几点：①并行分布处理；②高度鲁棒性和容错能力；③自组织及学习能力；④能充分逼近复杂的非线性关系.

人工神经网络的模型现在有数十种之多，在这里主要介绍应用较多的典型的神经网络模型——BP 神经网络.

9.1.2　BP 神经网络

1. 神经元的数学模型

在神经网络结构上，大量不同的神经元的轴突末梢可以到达同一个神经元的树突并形成大量突触. 来源不同的突触所释放的神经递质都可以对同一个神经元的膜电位变化产生作用. 因此，在树突上，神经元可以对不同来源的输入信息进行综合. 这就是神经元对信息的空间综合特性.

对于来自同一个突触的信息，神经元可以对不同时间传入的信息进行综合. 故神经元对信息有时间综合特性.

从神经元的特性和功能可以知道，神经元是一个多输入单输出的信息处理单元，而且，它对信息的处理是非线性的. 根据神经元的特性和功能，可以把神经元抽象为一个简单的数学模型. 工程上用的人工神经元模型如图 9.1 所示.

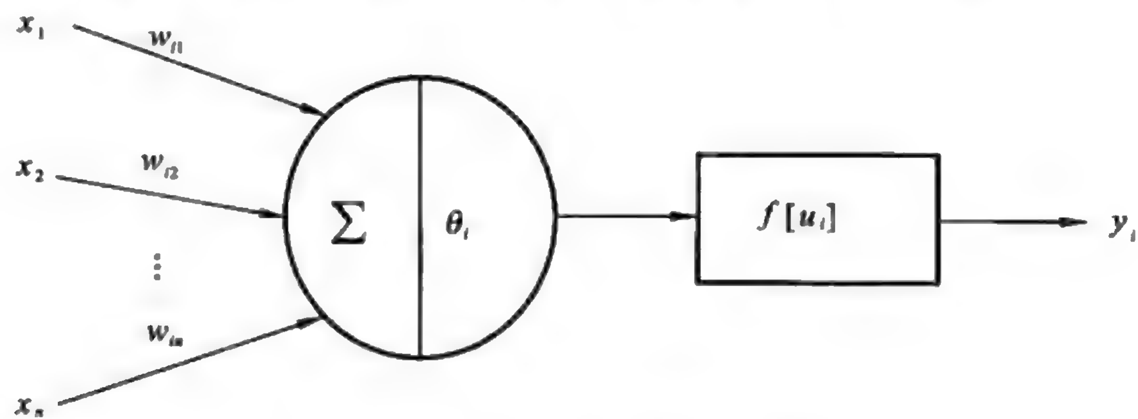
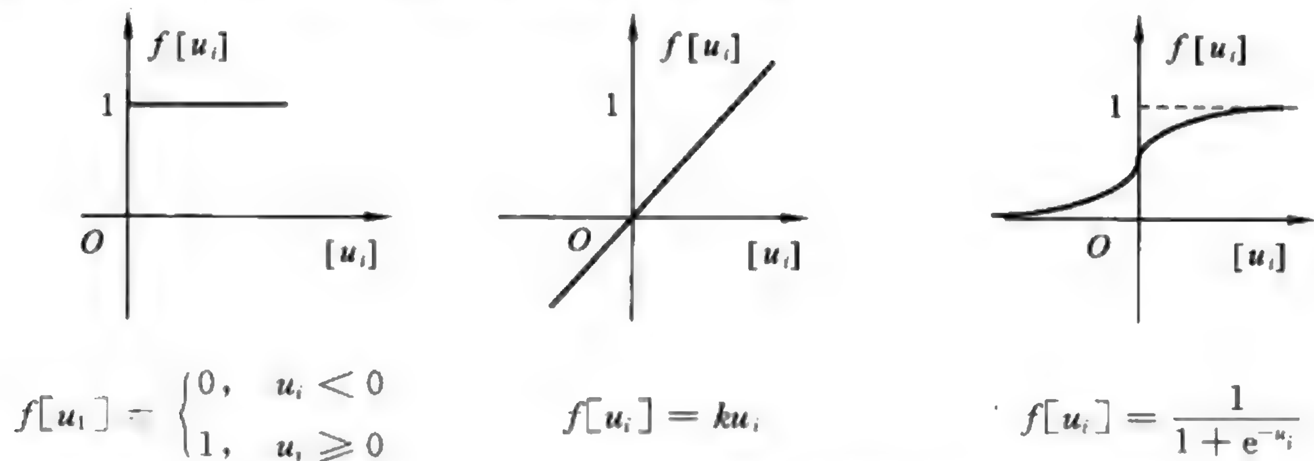


图 9.1　神经元数学模型示意图

图 9.1 中，$x_1,x_2,\cdots,x_n$ 是神经元的输入，也就是来自上一级神经元的信息；θ_i 表示神经元的阈值；y_i 表示神经元的输出；$f[u_i]$表示激发函数，它决定了在输入 x_1，$x_2,\cdots,x_n$ 的共同作用下，达到阈值 θ_i 时以何种形式输出.

图 9.2 列出了几种典型的激发函数的图像：



$f[u_i]=\begin{cases}0, & u_i<0\\ 1, & u_i\geqslant 0\end{cases}$　　$f[u_i]=ku_i$　　$f[u_i]=\dfrac{1}{1+e^{-u_i}}$

图 9.2　几种典型的激发函数

其中，$u_i=\sum\limits_j w_{ij}x_j-\theta_i$，所以

$$y_i=f[u_i]=f\Big(\sum_j w_{ij}x_j-\theta_i\Big) \tag{9.1.1}$$

(9.1.1) 式即为单个神经元的完整的数学模型表达式.

2. BP 神经网络的数学模型

BP 网络是反向传播(back propagation) 网络. 它是一种多层前向网络，采用最小均方误差学习方式. 这是一种最广泛应用的网络，可用于数据挖掘. BP 网络需有教师训练.

BP 算法的执行步骤　在反向传播算法应用于前馈多层网络时，采用 Sigmoid 为

激发函数时，可用下列步骤对网络的权系数 w_{ij} 进行递归求取. 注意对于每层有 n 个神经元的时候，对于第 k 层的第 i 个神经元，则有 n 个权系数 $w_{i1}, w_{i2}, \cdots, w_{jn}$，另外取多一个 w_{jn+1} 用于表示阈值 θ_i；并且在输入样本 $\boldsymbol{x}$ 时，取 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n, 1)$.

第一步 对权系数 w_{ij} 置初值. 对各层的权系数 w_{ij} 置一个较小的非零随机数，但其中 $w_{jn+1} = -\theta_i$.

第二步 输入一个样本 $\boldsymbol{x} = (x_1, x_2, \cdots x_n, 1)$，以及对应期望输出 $\boldsymbol{y} = (y_1, y_2, \cdots, y_n, 1)$.

第三步 计算各层的输出，对于第 k 层第 i 个神经元的输出 x_{ik}，有

$$y_i^k = f[u_i^k]$$

其中，

$$u_i^k = \sum_j w_{ij} x_j^{k-1} - \theta_i^k$$

式中，$x_{n+1}^{k-1} = 1, w_{i(n+1)} = -\theta$.

第四步 求各层的学习误差 d_i^k，对于输出层有 $k = m$，有

$$d_i^m = x_i^m(1 - x_i^m)(x_i^m - y_i^m)$$

对于其他各层，有

$$d_i^k = x_i^k(1 - x_i^k)\left(\sum_j w_{ij} x_j^{k-1} - \theta_i^k\right)$$

第五步 修正权系数 w_{ij} 和阈值 θ_i，有

$$w_{ij}(t+1) = w_{ij}(t) - \eta d_i^k x_j^{k-1}$$

第六步 当求出了各层各个权系数之后，可按给定品质指标判别是否满足要求. 如果未满足要求，则返回第三步继续执行；如果满足要求，则算法结束.

9.1.3 案例分析

例 9.1 1981 年生物学家格若根(W. Grogan) 和维什(W. Wirth) 发现了两类飞蠓. 他们测量了这两类飞蠓每个个体的翼长和触角长，数据见表 9.1.

表 9.1 两种飞蠓的特征数据

类别	飞蠓的特征数据								
Apf	触角长 /cm	1.14	1.18	1.20	1.24	1.26	1.28	1.30	1.36
	翼长 /cm	1.78	1.96	1.86	1.72	2.00	2.00	1.96	1.74
Af	触角长 /cm	1.38	1.38	1.38	1.40	1.48	1.54	1.56	
	翼长 /cm	1.64	1.82	1.90	1.70	1.82	1.82	2.08	

问：如果抓到三只新的飞蠓，它们的触角长和翼长分别为(1.24,1.80)，(1.28,1.84)，(1.40,2.04)，它们应分别属于哪一个种类?

分析建模 将问题视为一个系统，飞蠓的数据作为输入，飞蠓的类型作为输出，研究输入与输出的关系. 输入数据有 15 个，即 $p = 1, 2, \cdots, 15$；$j = 1, 2$；对应 15

个输出.

建立一个只有输入层与输出层的神经网络模型，输入层采用图 9.2 中第一种激发函数：

$$f[u_i]=\begin{cases}0, & u_i<0\\ 1, & u_i\geqslant 0\end{cases}$$

输出层采用图 9.2 中第二种激发函数：

$$f[u_i]=ku_i$$

为便于计算机处理，可以将符号数字化. 将 Apf 类记为 1，Af 类记为 0. 利用 MATLAB 中的 ANN 工具箱函数，编写如下 M- 文件：

```
p=[1.78 1.96 1.86 1.72 2.00 2.00 1.96 1.74 1.64 1.82 1.90 1.70 1.82 1.82 2.08;
   1.14 1.18 1.20 1.24 1.26 1.28 1.30 1.36 1.38 1.38 1.38 1.40 1.48 1.54 1.56];
%输入两种飞蠓的参数
t=[1 1 1 0 1 1 1 0 0 0 0 0 0 0 0];  %两种飞蠓的类别
plot(p(1,find(t>0.5)),p(2,find(t>0.5)),'o');hold on;
plot(p(1,find(t<=0.5)),p(2,find(t<=0.5)),'*');  %画出两种飞蠓的分布图
net=newff(minmax(p),[2,1],{'tansig','purelin'});
%建立一个具有两层的神经网络
net.trainParam.show=50;
net.trainParam.epochs=300;
net.trainParam.goal=1e-2;           %设置训练参数
net=train(net,p,t);                 %对网络进行训练估计出参数
pp=[1.801.842.04;1.241.281.40];     % 输入需要判别的三只飞蠓参数
y=sim(net,pp);                      %利用已训练好的网络识别三只飞蠓
plot(pp(1,find(y>0.5)),pp(2,find(y>0.5)),'ro');
plot(pp(1,find(y<=0.5)),pp(2,find(y<=0.5)),'r*');  %画出三只飞蠓的分布图
plot([1.72.1],[1.11.5],'k')  %画出判别直线
```

直接运行以上的脚本文件可以得到如下的运行结果：$y=(0.9850, 0.9936, 0.0286)$，说明三只飞蠓分别属于 Apf，Apf，Af 类. 可以直观地从图 9.3 看出.

结果分析 由于神经网络的初始权值由计算机随机给出，故每次运行的结果有一些差异，下面的一个答案是作者运行的结果：$y=(0.9850, 0.9936, 0.0286)$，说明三只飞蠓可以认为(1.24，1.80)，(1.28，1.84)，(1.40，2.04) 分别属于 Apf，Apf，Af 类.

值得注意的是：对于一般的预测方法而言，已经证实如果对于已知数据拟合过度，则其预测能力(或称为推广能力) 则会降低，此所谓过拟合现象. 本例中由于翼长和触角长的数据介于 0 ～ 3 之间，从而选择训练误差为 10^{-2} 作为停机条件是比较合适的.

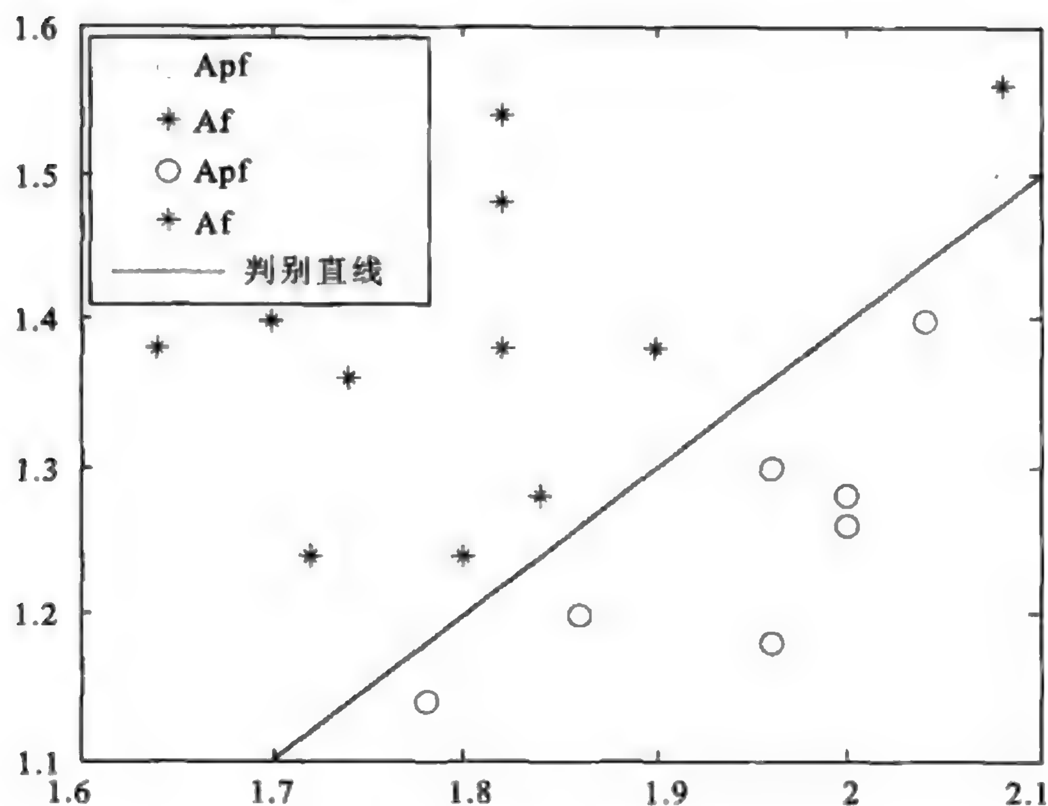


图 9.3　两种飞蠓的分类

9.1.4　方法评论

神经网络实现了一个从输入到输出的映射功能，而数学理论已证明它具有实现任何复杂非线性映射的功能. 这使得它特别适合于求解内部机制复杂的问题；能通过学习带正确答案的实例集自动提取“合理的”求解规则，即具有自学习能力；具有一定的推广、概括能力.

但是由于 BP 算法本质上为梯度下降法，而它所要优化的目标函数又非常复杂，因此，必然会出现“锯齿形现象”，这使得 BP 算法低效；而且从数学角度看，BP 算法为一种局部搜索的优化方法，但它要解决的问题为求解复杂非线性函数的全局极值，因此，算法很有可能陷入局部极值，使训练失败；神经网络的预测能力（也称泛化能力、推广能力）与训练能力（也称逼近能力、学习能力）的矛盾. 一般情况下，训练能力差时，预测能力也差，并且一定程度上，随训练能力地提高，预测能力也提高. 但这种趋势有一个极限，当达到此极限时，随训练能力的提高，预测能力反而下降，即出现所谓“过拟合”现象. 此时，神经网络学习了过多的样本细节，而不能反映样本的规律.

9.2　计算机仿真

计算机仿真是一种非实物仿真方法，是在研究系统过程中根据相似原理，利用计算机来逼真模仿研究对象，对研究对象进行数学描述，建立模型，这个模型包含所研究的系统的主要特点. 通过这个实验模型的运行，获得所要研究系统的必要信息，了解系统随时间变化的行为或特性，来评价或预测一个系统的行为效果，为决策提供信

息的一种方法.它是解决较复杂的实际问题的一条有效途径.可用于解决鼠疫的检测和预报,三峡的安全、生态,公交车的调度,航空管理,经营投资,道路的修建,通信网络服务,电梯系统服务等问题.

运用计算机仿真可解决以下5类问题:

(1) 难以用数学公式表示的系统,或者没有建立和求解数学模型的有效方法.

(2) 虽然可以用解析的方法解决问题,但数学的分析与计算过于复杂,这时计算机仿真可能提供简单可行的求解方法.

(3) 希望能在较短的时间内观察到系统发展的全过程,以估计某些参数对系统行为的影响.

(4) 难以在实际环境中进行实验和观察时,计算机仿真是唯一可行的方法,如太空飞行的研究.

(5) 需要对系统或过程进行长期运行比较,从大量方案中寻找最优方案.

计算机仿真在计算机中运行实现,不怕破坏,易修改,可重用,安全、经济,不受外界条件和场地空间的限制.

仿真分为静态仿真和动态仿真.动态仿真可分为连续系统仿真和离散系统仿真.离散系统是指状态变量只在某个离散时间点集合上发生变化的系统,如电梯系统服务、排队系统、通信网络服务的仿真等.连续系统是指状态变量随时间连续改变的系统,如传染病的检测和预报等.

仿真系统必须设置一个仿真时钟将时间从一个时刻向另一个时刻推进,并且可随时反映系统时间的当前值.模拟时间推进方式有两种:时间步长法和事件步长法.模拟离散系统常用事件步长法,连续系统常用时间步长法(也称固定增量推进法或步进式推进).

本节将介绍离散系统仿真、连续系统仿真、时间步长法、事件步长法和蒙特卡洛模拟.

9.2.1 准备知识:随机数的产生

由于仿真研究的实际系统要受到多种随机因素的作用和影响,在仿真过程中必须处理大量的随机因素.要解决这一个问题的前提是要确定随机变量的类型和选择合适的随机数产生的方法.

对随机现象进行模拟,实质上是要给出随机变量的模拟,也就是说要利用计算机随机产生一系列数值,使它们服从一定的概率分布,则称这些数值为随机数.

最常用的是(0,1)区间内均匀分布的随机数.其他分布的随机数可利用均匀分布的随机数产生.

1. 产生模拟随机数的计算机命令

在MATLAB软件中,可以直接产生满足各种分布的随机数,命令如下:

(1) 产生 $m\times n$ 阶(0,1)均匀分布的随机数矩阵:rand(m,n);产生一个(0,1)均匀分布的随机数:rand.

(2) 产生 $m\times n$ 阶$[a,b]$均匀分布 $U[a,b]$的随机数矩阵:unifrnd(a,b,m,n);产生一个$[a,b]$均匀分布的随机数:unifrnd(a,b).

(3) 产生 $m\times n$ 阶正态分布 $N(0,1)$标准正态分布随机数矩阵:randn(m,n);产生 $N(\mu,\sigma^2)$的正态分布的随机数:normrnd(μ,σ);产生 $m\times n$ 阶 $N(\mu,\sigma^2)$的正态分布随机数矩阵:normrnd(μ,σ,m,n).

(4) 产生 $m\times n$ 阶期望值为 μ 的指数分布随机数矩阵:exprnd(μ,m,n).

(5) 产生 $m\times n$ 阶参数为 λ 的泊松分布随机数矩阵:poissrnd(λ,m,n).

2. 案例分析

例 9.2 产生 5×3 阶(0,1)均匀分布的随机数矩阵;产生 1×7 阶[1,32]均匀分布的随机数矩阵;产生 1×7 阶 $N(70,25)$的正态分布随机数矩阵.

分析 由产生模拟随机数的计算机命令,很容易编写程序求解.

程序及结果:

```
>>rand(5,3)
 ans=
          0.2311    0.4565    0.7919
          0.6068    0.0185    0.9218
          0.4860    0.8214    0.7382
          0.8913    0.4447    0.1763
          0.7621    0.6154    0.4057
>>unifrnd(1,32,1,7)
 ans=1.3057   5.3056   7.2857   7.1604  19.7176   9.4378   7.1632
>>normrnd(70,25,1,7)
 ans=44.7342  85.3616  82.6935  112.3107  84.7821  53.9101  79.5084
```

注 由于随机性本质,同样的指令生成的随机结果会不同.

例 9.3 顾客到达某商店的间隔时间服从参数为 0.1 的指数分布,可以用 exprnd(10)模拟.该商店在单位时间内到达的顾客数服从参数为 0.1 的泊松分布,可以用 poissrnd(0.1)模拟.(这两个实际上是等价的)

分析 顾客到达某商店的间隔时间服从参数为 0.1 的指数分布指两个顾客到达商店的平均间隔时间是 10 个单位时间,即平均 10 个单位时间到达 1 个顾客.

该商店在单位时间内到达的顾客数服从参数为 0.1 的泊松分布指一个单位时间内平均到达 0.1 个顾客.

程序及结果:

(1) exprnd(10)

```
ans=2.0491
```

(2) poissrnd(0.1)

```
ans=1
```

例 9.4 敌空战部队对我方港口进行空袭,其达到规律服从泊松分布,平均每分钟达到 4 架飞机.

(1) 模拟敌机在 3 min 内达到目标区域的数量,以及在第 1,2,3 分钟内各达到几架飞机;

(2) 模拟在 3 min 分钟内每架飞机的到达时刻.

分析 (1) 敌机到达规律服从泊松分布,分别对每分钟敌机到达数量进行仿真.

(2) 由例 9.3 敌机到达规律服从泊松分布等价于敌机到达港口的间隔时间服从参数为 1/4 的指数分布,由指数分布,仿真出每架飞机的到达时刻.

(1) MATLAB 程序如下:

```
n1=poissrnd(4)   %模拟第 1 min 到达架数
n2=poissrnd(4)   %模拟第 2 min 到达架数
n3=poissrnd(4)   %模拟第 3 min 到达架数
n=n1+n2+n3     %3min 内到达架数之和
```

(2) MATLAB 程序如下:

```
clear
t=0;
j=0;  %到达的飞机数
while t<3
     j=j+1
     t=t+exprnd(1/4)
end
```

运行结果如下:

(1) n1=3,n2=8,n3=5,n=16.

(2) j=1,t=0.0409;j=2,t=0.0580;j=3,t=0.1548;j=4,t=0.2242;j=5,t=0.2984;j=6,t=0.5324;j=7,t=0.6380;j=8,t=1.0792;j=9;t=1.1663;j=10,t=2.0281;j=11,t=2.3491;j=12,t=3.1179.

9.2.2 随机变量的模拟

利用均匀分布的随机数可以产生具有任意分布的随机变量的样本,从而可以对随机变量的取值情况进行模拟.

1. 离散型随机变量的模拟

设随机变量 X 的分布律为 $P\{X=x_i\}=p_i \ (i=1,2,\cdots)$,令

$$p^{(0)}=0,\qquad p^{(n)}=\sum p_i,\quad n=1,2,\cdots$$

将 $p^{(n)}$ 作为分点,将区间(0,1)分为一系列小区间$(p^{(n-1)},p^{(n)})$. 对于均匀的随机变量

$R \sim U(0,1)$,则有

$$P\{p^{(n-1)} < R \leqslant p^{(n)}\} = p^{(n)} - p^{(n-1)}, \quad n = 1,2\cdots$$

由此可知,事件$\{p^{(n-1)} < R \leqslant p^{(n)}\}$和事件$\{X = x_i\}$有相同的发生的概率. 因此我们可以用随机变量 R 落在小区间内的情况来模拟离散的随机变量 X 的取值情况.

例 9.5 随机变量 $X = \{0,1,2\}$ 表示每分钟到达银行柜台的人数,X 的分布列见表 9.2,试模拟 10 min 内顾客到达柜台的情况.

表 9.2 10 min 内顾客到达柜台的情况

X_k	0	1	2
p_k	0.4	0.3	0.3

分析 因为每分钟到达柜台的人数是随机的,所以可用计算机随机生成一组(0,1) 的数据,由 X 的概率分布情况,可认为随机数在(0,0.4) 范围内时没有顾客光顾,在[0.4,0.7) 时,有一个顾客光顾,在[0.7,1) 时,有两个顾客光顾.

MATLAB 程序如下:

```
r=rand(1,10);
      for i=1:10;
        if r(i)<0.4
          n(i)=0;
        elseif 0.4<=r(i)&r(i)<0.7
          n(i)=1;
        else n(i)=2;
       end;
      end
    r
    n
```

显示随机结果:

```
r=0.0579   0.3529   0.8132   0.0099   0.1389   0.2028   0.1987   0.6038
   0.2722   0.1988
n= 0   0   2   0   0   0   0   1   0   0
```

2. 连续型随机变量的模拟

具有给定分布的连续型随机变量可以利用在区间(0,1) 上均匀分布的随机数来模拟. 最常用的方法是逆变换法.

结论 若随机变量 Y 有连续的分布函数 $F(y)$,而 X 是区间(0,1) 上均匀分布的随机变量,令 $Z = F^{-1}(X)$,则 Z 与 Y 有相同的分布.

若已知 Y 的概率密度为 $f(y)$,有 $F(Y) = \int_{-\infty}^{y} f(y)\mathrm{d}y$,如果给定区间(0,1) 上均

匀分布的随机数 r_i，则具有给定分布的随机数 y_i 可由方程 $r_i=\int_{-\infty}^{y_i}f(y)\mathrm{d}y$ 解出.

例 9.5 模拟服从参数为 λ 的指数分布时，由 $r_i=\int_{-\infty}^{y_i}\lambda e^{-\lambda y}\mathrm{d}y=1-e^{-\lambda y_i}$ 可得 $y_i=-\frac{1}{\lambda}\ln(1-r_i)$，也可简化为 $y_i=-\frac{1}{\lambda}\ln r_i$.

9.2.3 时间步长法

时间步长法可用于随时间变化的情况的过程模拟，公交车调度、激光辐照转动充压圆柱壳体热力学效应等问题中均有广泛应用.

1. 理论介绍

时间步长法(也称固定增量推进法或步进式推进)按照时间流逝的顺序，一步步对系统的活动进行仿真. 仿真过程分为许多相等的时间间隔，时间步长的长度可根据实际问题分别取为秒、分、小时、天等. 仿真时钟按时间步长等距推进，每次推进都要扫描系统中所有的活动，按照预定的计划和目标进行分析、计算，记录系统状态的变化，直到满足某个终止条件为止. 时间步长法的流程图如图 9.4 所示.

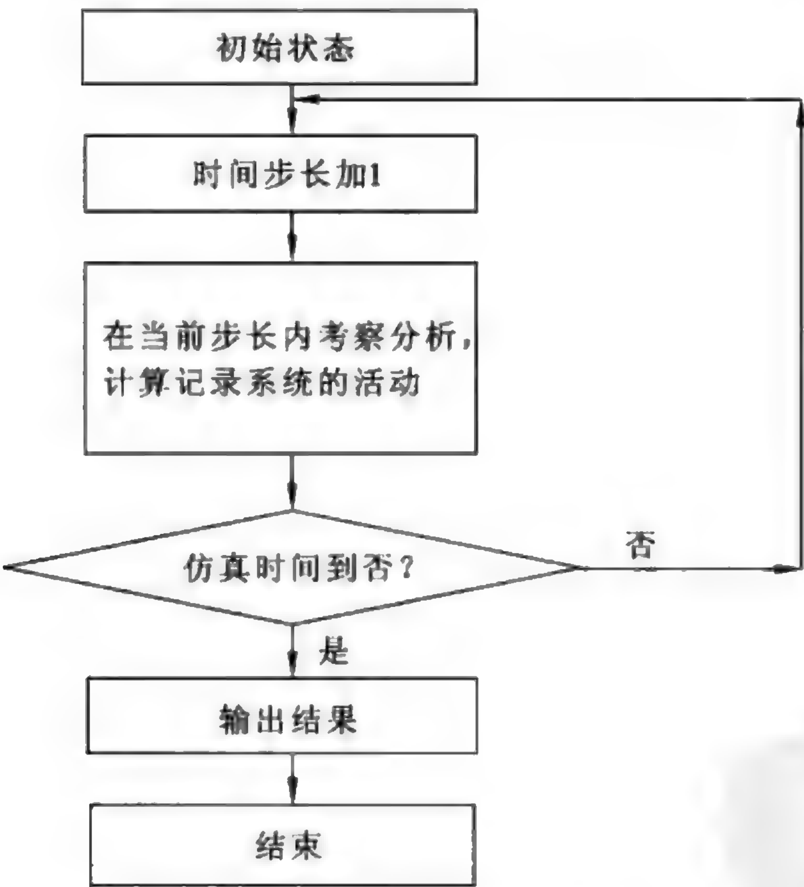


图 9.4 时间步长法流程图

2. 案例分析

例 9.6 某水池有 2000 m³ 水，其中含盐 2 kg，以 6 m³/min 的速率向水池内注入含盐为 0.5 kg/m³ 的盐水，同时又以 4 m³/min 的速率从水池流出搅拌均匀的盐水. 试用计算机仿真该水池内盐水的变化过程，并每隔 10 min 计算水池中水的体积、含

盐量和含盐率. 欲使池中盐水的含盐率达到 0.2 kg/m^3,需经过多长时间?

分析 这是一个连续系统,首先要将系统离散化,在一些离散点上进行考察,这些离散点的间隔就是时间步长. 可以取步长为 1 min,即隔 1 min 考察一次系统的状态,并相应地记录和分析. 在注入和流出活动的作用下,池中水的体积与含盐量、含盐率均随时间变化,初始时刻含盐率为 0.001 kg/m^3,以后每分钟注入含盐率为 0.5 kg/m^3 的水 6 m^3,流出混合均匀的盐水 4 m^3,当池中水的含盐率达到 0.2 kg/m^3 时,仿真过程结束.

记 T 时刻水的体积为 w(m^3),水的含盐量为 s(kg),水的含盐率为 $r = s/w$(kg/m^3),每隔 1 min 池水的动态变化过程如下:每分钟水的体积增加 $6-4=2$(m^3);每分钟向池内注入盐 $6\times 0.5=3$(kg);每分钟向池外流出盐 $4\times r$(kg);每分钟池内增加盐 $3-4\times r$(kg).

MATLAB 程序如下:

```
clear
h=1;  %时间步长为 1
s0=2;  %初始含盐 2kg
w0=2000;  %初始水池有水 2000m³
r0=s0/w0;  %初始浓度
s(1)=s0+0.5*6*h-4*h*r0;  %1 min 后的含盐量
w(1)=w0+2*h;  %1 min 后水池中的盐水体积
r(1)=s(1)/w(1);  %1 min 后的浓度
t(1)=h;
y(1)=(2000000+ 3000000*h+3000*h^2+h^3)/(1000+h)^2;
for i= 2:200
    t(i)=i*h;
    s(i)=s(i-1)+0.5*6*h- 4*h*r(i-1);  %第 i 步后的含盐量
    w(i)=w(i-1)+2*h;  %第 i 步后的盐水体积
    r(i)=s(i)/w(i);  %第 i 步后的盐水浓度
    y(i)=(2000000+3000000*t(i)+ 3000*t(i)^2+t(i)^3)/(1000+t(i))^2;
    m=floor(i/10);
    if i/10-m<0.1
          tm(m)=m;
          wm(m)=w(i);
          sm(m)=s(i);
          rm(m)=r(i);
    end
    if r(i)>0.2  %若第 i 步后的盐水浓度大于 0.2
          t02=i*h;
```

```
            r02=r(i);
            break
        end
      end
    [t02,r02]
    [10*tm',sm',rm']  % '表示转置
    subplot(1,2,1),plot(t,s,'blue');
    hold on
    subplot(1,2,2),plot(t,y,'red');
```

本例还可以用微分方程建立数学模型，并求出它的解析解，这个解析解就是问题的精确解. 有兴趣的读者可以按照这个思路求出该问题的精确解，考察相应时刻精确解与仿真解的差异，还可以进一步调整仿真过程的时间步长，通过与精确解的比较来研究时间步长的大小对仿真精度的影响.

9.2.4 事件步长法

事件步长法常用于排队系统中，排队系统是随机系统的一个大类，包括各种交通系统、电话系统、加工系统等，是离散事件系统仿真应用的经典系统. 这些系统中被服务者的到来时间、服务时间的长短以及系统中被服务者数量等都不是一个确定值，而是一个随机值，随机性是这类系统的固有属性，所以被称为随机服务系统.

事件步长法是以事件发生的时间为增量，按照时间的进展，一步一步地对系统的行为进行仿真，直到预定的时间结束为止. 其过程是：设置仿真时钟初值为 0，跳到第一个事件发生的时刻，计算系统的状态，产生未来事件并加入到队列中去；跳到下一事件，计算系统的状态；……重复这一过程直到满足某个终止条件为止.

例 9.7 (收款台前的排队过程)假设：

(1) 顾客到达收款台是随机的，平均时间间隔为 0.5 min，即间隔时间服从 $\lambda=2$ 的指数分布.

(2) 对不同的顾客收款和装袋的时间服从正态分布 $N(1,1/3)$.

试模拟 20 位顾客到收款台前的排队情况，我们关心的问题是每个顾客的平均等待时间、队长及服务员的工作效率.

分析 单服务台结构的排队系统有两类原发事件即到来和离去，顾客到来的后继事件是顾客接受服务，顾客离去的后继事件是服务台寻找服务，这 4 类事件各自的子程序框图如图 9.5 所示.

假设：$t(i)$为第 i 位顾客到达时刻；$t_2(i)$为第 i 位顾客受到的服务时间(随机变量)；$T(i)$为第 i 位顾客离去时刻.

将第 i 位顾客到达作为第 i 件事发生：$t(i+1)-t(i)=r(i)$(随机变量)；平衡关系：当 $t(i+1)\geq T(i)$时，$T(i+1)=t(i+1)+t_2(i+1)$；否则，$T(i+1)=T(i)+t_2(i+1)$.

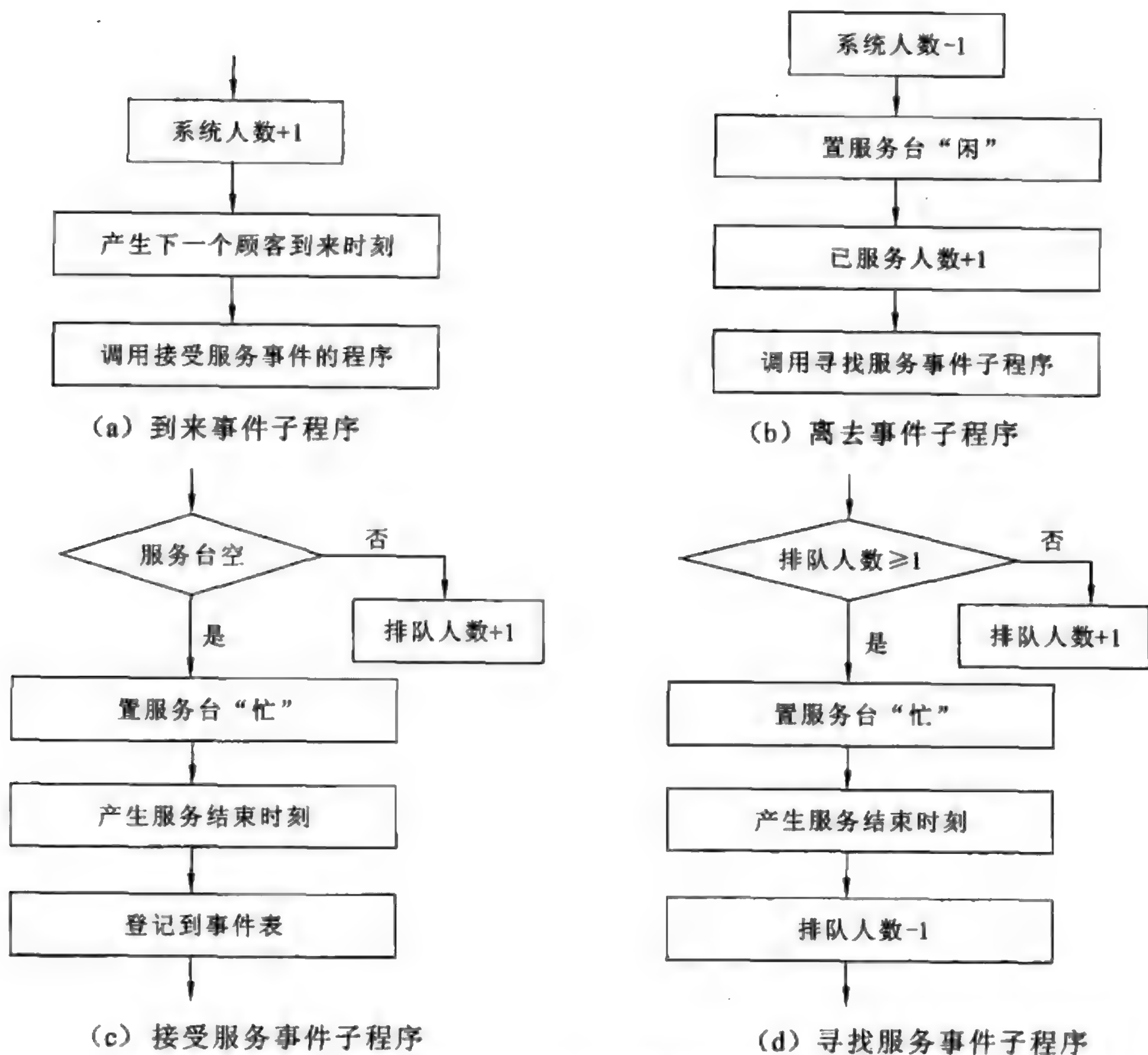


图 9.5 服务台事件流程图

MATLAB 程序如下：

```
clear
t=zeros(1,21);  %每位顾客到达时间
T=zeros(1,21);  %每位顾客离去时间
w=zeros(1,21);  %顾客等待时间累加
ww=zeros(1,21);  %收款台空闲时间累加
r=exprnd(2,1,21);  %服从指数分布的随机数
t2=normrnd(1,1/3,1,21);  %服从正态分布的随机数
for i=1:1:20
    t(i+1)=t(i)+r(i);
    if t(i+1)>=T(i);  %不需要排队,即第 i+1 个到来时候第 i 个已经服务完成
        w(i+1)=w(i);T(i+1)=t(i+1)+t2(i+1);
        ww(i+1)=t(i+1)-T(i)+ww(i);
```

```
        else w(i+1)=T(i)-t(i+1)+w(i);
            T(i+1)=T(i)+t2(i+1);ww(i+1)=ww(i);
        end;
    end;
    b=[t',T',w',ww'];
    [brow,bcol]=size(b);  %求队长,brow表示行数,bcol表示列数
    b=[b,zeros(brow,1)];  %zero生成brow行,1列的零矩阵
    for j=2:brow
        l=0;  %队列长度
        if j-l-1>0
            while b(j,1)<=b(j-l-1,2)
                l=l+1;
            end;
          b(j,bcol+1)=l;
      end;
    end;
    b
    g1=w(end)/20  %平均等待时间
    g2=sum(T-t)/20  %平均逗留时间
    g3= T(end)/20  %平均每分钟服务的顾客人数
```

运行结果如下：

```
g1=0.3778,g2=1.4340,g3=1.8288.
```

所以每个顾客的平均等待时间为 0.3778 min,平均每个顾客的逗留时间为 1.4340 min及平均每分钟服务的顾客人数为 1.8288 个.

9.2.5 蒙特卡罗模拟

蒙特卡罗(Monte Carlo)方法能够帮助人们从数学上表述物理、化学、工程、经济学以及环境动力学中一些非常复杂的相互作用,真实地模拟实际情况,解决问题与实际非常符合,可以得到很圆满的结果.

1. 理论介绍

蒙特卡罗模拟亦称为随机模拟(random simulation)方法,有时也称为随机抽样技术、统计试验方法,简称 M-C 模拟. M-C 模拟是静态模拟方法. 此方法对研究的系统进行随机观察抽样,通过对样本值的观察统计,求得所研究系统的某些参数. 其基本思想是:当所求解问题是某种随机事件出现的概率,或者是某个随机变量的期望值时,通过某种"实验"的方法,以这种事件出现的频率估计这一随机事件的概率,或者得到这个随机变量的某些数字特征,并将其作为问题的解.

2. 案例分析

例 9.8 用蒙特卡罗模拟求圆周率 π 的估计值.

在概率论中,蒲丰投针是用统计试验求圆周率 π 的典型方法,这里给出另一个求圆周率 π 的概率模型.

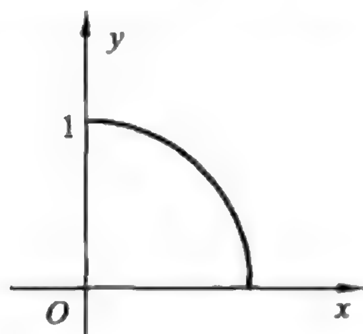


图 9.6 示意图

分析 如图 9.6 所示,考虑边长为 1 的正方形,以坐标原点为圆心,1 为半径在正方形内画出一条 1/4 圆弧. 设二维随机变量 (X,Y) 在正方形服从均匀分布,则 (X,Y) 落在 1/4 圆内的概率为 $P\{X^2+Y^2\leqslant 1\}=\pi/4$.

现产生 n 对二维随机点 (x_i,y_i),x_i,y_i 是区间 $(0,1)$ 上均匀分布的随机数,若其中有 k 对满足 $x_i^2+y_i^2\leqslant 1$,即相当于做 n 次投点试验,其中有 k 次落在 1/4 圆内. 计算落入 1/4圆内的频率为 k/n,由大数定律,事件发生的频率依概率收敛于发生的概率,可得圆周率 π 的估计值为 $4k/n$. 并且当试验次数越多时,估计值的精度也越来越高.

MATLAB 程序如下:

```
clear
n=50000
X=rand(n,1);
Y=rand(n,1);
k=0;
for i=1:n;
     if X(i)^2+ Y(i)^2<=1
     k=k+1;
    end
end
4*k/n
```

运行结果如下:

```
ans=3.1510
```

有兴趣的读者可以增加程序中投点的试验次数 n,从而提高精度.

9.2.6 应用举例

例 9.9 (库存问题)在物资的供应过程中,由于到货与销售不可能做到同步同量,故总要保持一定的库存储备. 如果库存过多,就会造成积压浪费以及保管费的上升;如果库存过少,会造成缺货. 如何选择库存和订货策略,就是一个需要研究的问题. 库存问题有多种类型,一般都比较复杂,下面讨论一种简单的情形.

某电动车行的仓库管理人员采取一种简单的订货策略,当库存量降低到 P 辆电动车时就向厂家订货,每次订货 Q 辆,如果某一天的需求量超过了库存量,商店就有

销售损失和信誉损失，但如果库存量过多，会导致资金积压和保管费增加. 若现在已有如表 9.3 所示的两种库存策略，试比较选择一种策略以使总费用最少.

表 9.3　订货方案

方案	重新订货点 P/辆	重新订货量 Q/辆
方案 1	125	150
方案 2	150	250

这个问题的已知条件是：

(1) 从发出订货到收到货物需隔 3 天.

(2) 每辆电动车保管费为 0.50 元/天，每辆电动车的缺货损失为 1.60 元/天，每次的订货费为 75 元.

(3) 每天电动车需求量是 0 到 99 之间均匀分布的随机数.

(4) 原始库存为 110 辆，并假设第一天没有发出订货.

分析　这一问题用解析法讨论比较麻烦，但用计算机按天仿真仓库货物的变动情况却很方便. 我们以 30 天为例，依次对这两种方案进行仿真，最后比较各方案的总费用，从而就可以做出决策.

计算机仿真时的工作流程是早上到货、全天销售、晚上订货，输入一些常数和初始数据后，以一天为时间步长进行仿真. 首先检查这一天是否为预定到货日期，如果是，则原有库存量加 Q，并把预定到货量清为零；如果不是，则库存量不变. 接着仿真随机需求量，这可用计算机语言中的随机函数得到. 若库存量大于需求量，则新的库存量减去需求量；反之，则新库存量变为零，并且要在总费用上加缺货损失，然后检查实际库存量加上预定到货量是否小于重新订货点 P，如果是，则需要重新订货，这时就加一次订货费. 如此重复运行 30 天，即可得所需费用总值. 由此比较这两种方案的总费用，可以得到最好的方案.

由于程序源码过长，读者请登陆华中数学建模网 http://www.shumo.cn/大学数学实验栏目下载学习本程序.

运行结果如下：

```
cost=2308          2908
mincost=2308
```

两种方案的花费分别为 2308，2908，第一种方案花费小，所以第一种方案好.

例 9.10　(赶火车过程仿真)一列火车从 A 站经过 B 站开往 C 站，某人每天赶往 B 站乘这趟火车. 已知火车从 A 站到 B 站运行时间为均值 30 min、标准差为 2 min 的正态随机变量. 火车大约在下午 1 点离开 A 站，离开时刻的频率分布见表 9.4. 这个人到达 B 站时的频率分布见表 9.5. 用计算机仿真火车开出、火车到达 B 站、这个人到达

B 站情况，并给出能赶上火车的仿真结果.

表 9.4 离开时刻的频率分布

出发时刻 T	1:00	1:05	1:10
频率	0.7	0.2	0.1

表 9.5 人到达 B 站时的频率分布

到达时刻 T	1:28	1:30	1:32	1:34
频率	0.3	0.4	0.2	0.1

分析 引入以下变量：T_1 为火车从 A 站开出的时刻；T_2 为火车从 A 站运行到 B 站所需要的时间；T_3 为此人到达 B 站的时刻. 则有表 9.6.

表 9.6 T_1, T_2 的频率分布

T_1/min	0	5	10	
p	0.7	0.2	0.1	
T_2/min	28	30	32	34
p	0.3	0.4	0.2	0.1

显然，这位旅客要赶上火车的条件是 $T_3 < T_1 + T_2$，那么可以通过计算机模拟出 T_1, T_2, T_3 这三个时间，再检验是否满足 $T_3 < T_1 + T_2$. 如果满足，即能够赶上火车，若不然，则不能赶上火车.

开车时间的仿真程序：

```
s1=0;s2=0;s3=0;
x=rand(10000,1);
for i=1:10000
      if x(i)<0.7
          s1=s1+ 1;
      end
      if x(i)>0.9
          s3=s3+1;
      end
end
[s1/10000,1-s1/10000-s3/10000,s3/10000]
```

人到达时刻仿真程序：

```
s1=0;s2=0;s3=0;s4=0;
x=rand(10000,1);
```

```
for i=1:10000
    if x(i)<0.3
        s1=s1+1;
    elseif x(i)<0.7
        s2=s2+1;
    elseif x(i)<0.9
        s3=s3+1;
    else
        s4=s4+1;
    end
    end
end
[s1/10000,s2/10000,s3/10000,s4/10000]
```

火车运行时间的仿真程序：

```
x=randn(10000,1);
for i=1:10000
    y(i)=30+2*x(i);
end
```

赶上火车的仿真程序：

```
s=0;
x1=rand(10000,1);
x2=rand(10000,1);
x3=randn(10000,1);
for i=1:10000
    if x1(i)<0.7
        T1=0;
    elseif x1(i)<0.9
        T1=5;
    else
        T1=10;
    end
        T2=30+2* x3(i);
    if x2(i)<0.3
        T3=28;
    elseif x2(i)<0.7
        T3=30;
    elseif x2(i)<0.9
        T3=32;
```

```
        else
            T3=34;
        end
        if T3<T1+T2
            s=s+1;
        end
    end
    [s/10000]
```

三种开车时间的概率分别为 0.6957,0.2036,0.1007,人到达 B 站的三个时刻的概率分别为 0.3044,0.3919,0.2037,0.1000,人赶上火车的概率为 0.6349.

9.2.7 方法评论

计算机仿真的精度受到许多方面因素的影响,比较难以控制.因此,在能用解析方法求解时,通常不用计算机仿真.计算机仿真环境局限于一个单维的数字化信息空间,将主体和客体人为分割开来,严重影响了人类对丰富多彩的自然界的多种渠道的理解和把握.不过随着虚拟现实等技术的发展,情况将有望获得改善.

在科技发展一日千里的今天,计算机仿真方法的应用仍然是非常广泛并且不可替代的.展望将来,计算机仿真方法将在人工智能领域、军事领域、医学领域、社会领域有着广阔的应用前景.

9.3 灰色系统

灰色系统理论(grey system theory)的创立源于 20 世纪 80 年代.邓聚龙教授在 1981 年上海中-美控制系统学术会议上所作的"含未知数系统的控制问题"的学术报告中首次使用了"灰色系统"一词.1982 年,邓聚龙发表了"参数不完全系统的最小信息正定"、"灰色系统的控制问题"等系列论文,奠定了灰色系统理论的基础.他的论文在国际上引起了高度的重视,美国哈佛大学教授、《系统与控制通信》杂志主编布罗克特(Brockett)给予灰色系统理论高度评价,因而,众多的中青年学者加入到灰色系统理论的研究行列,积极探索灰色系统理论及其应用研究.

事实上,灰色系统的概念是由英国科学家艾什比(W. R. Ashby)所提出的"黑箱"(black box)概念发展演进而来,是自动控制和运筹学相结合的产物.艾什比利用黑箱来描述那些内部结构、特性、参数全部未知而只能从对象外部和对象运动的因果关系及输出输入关系来研究的一类事物.邓聚龙系统理论则主张从事物内部,从系统内部结构及参数去研究系统,以消除"黑箱"理论从外部研究事物而使已知信息不能充分发挥作用的弊端,因而,被认为是比"黑箱"理论更为准确的系统研究方法.所谓灰色系统是指部分信息已知而部分信息未知的系统,灰色系统理论所要考察和研究

的是对信息不完备的系统，通过已知信息来研究和预测未知领域从而达到了解整个系统的目的. 灰色系统理论与概率论、模糊数学一起并称为研究不确定性系统的三种常用方法，具有能够利用"少数据"建模寻求现实规律的良好特性，克服了资料不足或系统周期短的矛盾.

目前，灰色系统理论得到了极为广泛的应用，不仅成功地应用于工程控制、经济管理、社会系统、生态系统等领域，而且在复杂多变的农业系统，如在水利、气象、生物防治、农机决策、农业规划、农业经济等方面也取得了可喜的成就. 灰色系统理论在管理学、决策学、战略学、预测学、未来学、生命科学等领域展示了极为广泛的应用前景.

灰色系统 GM(1,1)模型是依据系统中已知的多种因素的综合资料，将此资料的时间序列按微分方程拟合去逼近上述时间序列所描述的动态过程，进而外推，达到预测的目的. 这种拟合得到的模型是时间序列的一阶微分方程，因此，简记为 GM(1,1) 模型.

9.3.1 理论介绍

1) 两个概念：累加生成数(AGO) 和累减生成数(IAGO)

(1) 累加生成数 1-AGO 指一次累加生成. 记原始序列为

$$X^{(0)} = \{x^{(0)}(1), x^{(0)}(2), \cdots, x^{(0)}(n)\}$$

一次累加生成序列为

$$X^{(1)} = \{x^{(1)}(1), x^{(1)}(2), \cdots, x^{(1)}(n)\}$$

其中，
$$x^{(1)}(k) = \sum_{i=0}^{k} x^{(0)}(i) = x^{(1)}(k-1) + x^{(0)}(k)$$

(2) 累减生成数(IAGO) 是累加生成的逆运算. 记原始序列为

$$X^{(1)} = \{x^{(1)}(1), x^{(1)}(2), \cdots, x^{(1)}(n)\}$$

一次累减生成序列为

$$X^{(0)} = \{x^{(0)}(1), x^{(0)}(2), \cdots, x^{(0)}(n)\}$$

其中，
$$x^{(0)}(k) = x^{(1)}(k) - x^{(1)}(k-1)$$

规定 $x^{(1)}(0) = 0$.

2) GM(1,1) 模型

符号的含义：表示一阶、一个变量的灰色系统模型. 令 $X^{(0)}$ 表示需要建模的序列，$X^{(1)}$ 为 $X^{(0)}$ 的 1-AGO 序列，则有

$$x^{(1)}(k) = \sum_{i=1}^{k} x^{(0)}(i)$$

定义 $Z^{(1)}$ 为 $X^{(1)}$ 的紧邻均值(MEAN) 生成序列：

$$z^{(1)}(k) = \frac{x^{(1)}(k) + x^{(1)}(k-1)}{2}$$

则可建立如下灰微分方程：

$$x^{(0)}(k)+az^{(1)}(k)=b$$

记 $\hat{\boldsymbol{\alpha}}=(a,b)^{\mathrm{T}}$，则灰微分方程的最小二乘估计参数列满足下式：

$$\hat{\boldsymbol{\alpha}}=(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{Y}_n$$

其中，

$$\boldsymbol{B}=\begin{bmatrix}-z^{(1)}(2) & 1\\ -z^{(1)}(3) & 1\\ \vdots & \vdots\\ -z^{(1)}(n) & 1\end{bmatrix},\quad \boldsymbol{Y}_n=\begin{bmatrix}x^{(1)}(2)\\ x^{(1)}(3)\\ \vdots\\ x^{(1)}(n)\end{bmatrix}$$

称 $\frac{\mathrm{d}x^{(1)}}{\mathrm{d}t}+ax^{(1)}=b$ 为灰微分方程 $x^{(0)}(k)+az^{(1)}(k)=b$ 的白化方程，也称为影子方程.

综上所述，则有

(1) 白化方程 $\frac{\mathrm{d}x^{(1)}}{\mathrm{d}t}+ax^{(1)}=b$ 的解也称为时间响应函数：

$$\hat{x}^{(1)}(t)=\left(x^{(1)}(0)-\frac{b}{a}\right)\mathrm{e}^{-at}+\frac{b}{a}$$

(2) GM(1,1) 灰色微分方程 $x^{(0)}(k)+az^{(1)}(k)=b$ 的时间响应序列为

$$\hat{x}^{(1)}(k+1)=[x^{(1)}(0)-\frac{b}{a}]\mathrm{e}^{-ak}+\frac{b}{a},\quad k=1,2,\cdots,n$$

(3) 取 $x^{(1)}(0)=x^{(0)}(1)$，则有

$$\hat{x}^{(1)}(k+1)=\left[x^{(0)}(1)-\frac{b}{a}\right]\mathrm{e}^{-ak}+\frac{b}{a},\quad k=1,2,\cdots,n$$

(4) 将值还原得到

$$\hat{x}^{(0)}(k+1)=\hat{x}^{(1)}(k+1)-\hat{x}^{(1)}(k)\tag{9.3.1}$$

(9.3.1) 式即为预测方程.

3) GM(1,1) 模型的检验

GM(1,1) 模型的检验分为三个部分：残差检验、关联度检验以及后验差检验.

(1) 残差检验：残差 $\varepsilon=|\hat{X}-X^{(0)}|$，相对误差 $\xi=\frac{\varepsilon}{X^{(0)}}$.

(2) 关联度检验：

$$\rho^{(i)}(k)=\frac{\min\limits_i\min\limits_k|x^{(0)}(k)-x^{(1)}(k)|+\lambda\max\limits_i\max\limits_k|x^{(0)}(k)-x^{(1)}(k)|}{|x^{(0)}(k)-x^{(1)}(k)|+\lambda\max\limits_i\max\limits_k|x^{(0)}(k)-x^{(1)}(k)|}$$

其中，$\lambda\in[0,1]$ 为分辨系数，一般取 $\lambda=0.5$.

但是武汉理工大学的吴涛、李必强与海军装备论证研究中心舰船所钱正芳等人

通过对关联度系数存在不合理下限分析，对不合理下限进行修正后，用于 GM(1,1) 模型的检验，得出关联度检验具有与残差检验和后验差检验不一致的结果，认为关联度不适合于 GM(1,1) 模型的检验.

(3) 后验差检验：

$$C = \frac{S_1}{S_2}$$

其中，s_1，s_2 分别为残差序列的均方差和原序列的均方差，

$$S_1 = \frac{\sqrt{\sum (\varepsilon_i - \bar{\varepsilon})^2}}{n-1}, \qquad \bar{\varepsilon} = \frac{\sum \varepsilon_i}{n}$$

$$S_2 = \frac{\sqrt{\sum [X^{(0)}(k) - \overline{X}^{(0)}]^2}}{n-1}, \qquad \overline{X}^{(0)} = \frac{\sum X^{(0)}(k)}{n}$$

查后验差检验判别参照表(表 9.7)，可以判断模型的精度.

表 9.7　后验差检验判别参照表

C	模型精度
< 0.35	优
< 0.5	合格
< 0.65	勉强合格
> 0.65	不及格

9.3.2　案例分析

例 9.10　由 1990 ～ 2002 年中国蔬菜产量，建立模型预测 2003 年中国蔬菜产量，并对预测结果做检验.

分析建模　给定原始时间 1990 ～ 2001 年资料序列 $X^{(0)}(k)$，对 $X^{(0)}(k)$ 生成 1-AGO 序列 $X^{(1)}(k)$，另外可得 $\boldsymbol{Y}_n$ 见表 9.8.

表 9.8　原始序列 $X^{(0)}(k)$、生成的 1-AGO 序列 $X^{(1)}(k)$ 以及 Y_n

k	1	2	3	…	10	11	12
$X^{(0)}$	19519	19578	19637	…	40514	42400	48337
$X^{(1)}$	19519	39097	58734	…	264605	307005	355342
Y_n	—	39097	58734	…	264605	307005	355342

其中，$X^0 = \mathrm{AGO}(X^1)$，即

$$x^{(1)}(k) = \sum_{i=1}^{k} x^{(0)}(i)$$

$$Y_n = [x^{(1)}(2), x^{(1)}(3), \cdots, x^{(1)}(12)]^T$$

对于上述的 $X^{(0)}(k)$ 的 GM(1,1) 参数按照下式进行最小二乘估计得

$$B = \begin{bmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ -z^{(1)}(4) & 1 \\ -z^{(1)}(5) & 1 \\ -z^{(1)}(6) & 1 \\ -z^{(1)}(7) & 1 \\ -z^{(1)}(8) & 1 \\ -z^{(1)}(9) & 1 \\ -z^{(1)}(10) & 1 \\ -z^{(1)}(11) & 1 \\ -z^{(1)}(12) & 1 \end{bmatrix} = \begin{bmatrix} -0.5[x^{(1)}(1)+x^{(1)}(2)] & 1 \\ -0.5[x^{(1)}(2)+x^{(1)}(3)] & 1 \\ -0.5[x^{(1)}(3)+x^{(1)}(4)] & 1 \\ -0.5[x^{(1)}(4)+x^{(1)}(5)] & 1 \\ -0.5[x^{(1)}(5)+x^{(1)}(6)] & 1 \\ -0.5[x^{(1)}(6)+x^{(1)}(7)] & 1 \\ -0.5[x^{(1)}(7)+x^{(1)}(8)] & 1 \\ -0.5[x^{(1)}(8)+x^{(1)}(9)] & 1 \\ -0.5[x^{(1)}(9)+x^{(1)}(10)] & 1 \\ -0.5[x^{(1)}(10)+x^{(1)}(11)] & 1 \\ -0.5[x^{(1)}(11)+x^{(1)}(12)] & 1 \end{bmatrix} = \begin{bmatrix} -29308.0 & 1 \\ -48915.5 & 1 \\ -68581.5 & 1 \\ -86730.0 & 1 \\ -107892.5 & 1 \\ -135943.5 & 1 \\ -168369.5 & 1 \\ -204848.5 & 1 \\ -244348.0 & 1 \\ -331173.5 & 1 \\ -5236.2 & 1 \end{bmatrix}$$

将 B, Y_n 代入辨识算式，有

$$\hat{a} = \begin{bmatrix} a \\ b \end{bmatrix} = (B^T B)^{-1} B^T Y_n = \begin{bmatrix} -0.1062105 \\ 13999.9 \end{bmatrix}$$

得 GM(1,1) 模型为

(1) 灰微分方程：$X^{(0)}(k) - 0.1062105 z^{(1)}(k) = 13999.9$

(2) 白化方程： $\dfrac{dx^{(1)}}{dt} - 0.1062105 z^{(1)}(k) = 13999.9$

(3) 白化方程的时间响应式：

$$\hat{x}^{(1)}(t+1) = \left[x^{(0)}(1) - \frac{b}{a}\right]e^{-at} + \frac{b}{a} = 151332.5e^{0.1062105t} - 131813.5$$

得还原方程为

$$\hat{x}^{(0)}(t+1) = 15248.968e^{0.1062105t}$$

利用 MATLAB 编程预测 2003 年中国蔬菜产量，并对预测结果做残差检验和后验差检验，程序如下：

```
function[X,c,error1,error2]=GM11(X0,k)
%建立函数[X,c,error1,error2]=GM11(X0,k)
%其中 X0 为输入序列,k 为预测长度,
%X 为预测输出序列,c 为后验差检验数,error1 为残差,error2 为相对误差
format long;
n=length(X0);
X1=[];
X1(1)=X0(1);
for i=2:n
```

```
        X1(i)=X1(i-1)+X0(i);  %计算累加生成序列
    end
    for i=1:n-1
        B(i,1)=-0.5*(X1(i)+X1(i+1));  %计算 B,Yn
        B(i,2)=1;
        Y(i)=X0(i+1);
    end
    alpha=(B'*B)^(-1)*B'*Y';  %作最小二乘估计
    a=alpha(1,1);
    b=alpha(2,1);
    d=b/a;  %计算时间响应函数参数
    c=X1(1)-d;
    X2(1)=X0(1);
    X(1)=X0(1);
    for i=1:n-1
        X2(i+1)=c*exp(-a*i)+d;
        X(i+1)=X2(i+1)-X2(i);  %计算预测序列
    end
    for i=(n+1):(n+k)
        X2(i)=c*exp(-a*(i-1))+d;  %计算预测序列
        X(i)=X2(i)-X2(i-1);
    end
    for i=1:n
        error(i)=X(i)-X0(i);
        error1(i)=abs(error(i));  %计算残差
        error2(i)=error1(i)/X0(i);  %计算相对误差
    end
    c=std(error1)/std(X0);  %计算后验差检验数
```

输入命令：

```
X0=[19519,19578,19637,19695,16602,25723,30379,34473,38485,40514,42400,48337];
k=1;
[X,c,error1,error2]=GM11(X0,k)
plot(1991:2002,X0,'g*-')
holdon
plot(1991:2003,X)
```

运行结果见表 9.9.

表 9.9 1990～2002 年 GM(1,1)灰色系统预测值与实际值比较

年份	预测值/10^4t	实际值/10^4t	残差 $\varepsilon/10^4$t	相对误差 ξ/%
1991	16957.69	19578	2620.3	13.38
1992	18857.91	19637	779.09	3.967
1993	20971.05	19695	−1276	−6.48
1994	23320.96	16602	−6719	−40.5
1995	25934.29	25723	−211.3	−0.82
1996	28840.3	30379	1538.7	5.065
1997	32072.06	34473	2400.9	6.965
1998	35665.88	38485	2819.1	7.325
1999	39662.47	40514	851.53	2.102
2000	44106.89	42400	−1707	−4.03
2001	49049.32	48337	−712.3	−1.47

画出预测值与实际值的变化曲线如图 9.7 所示.

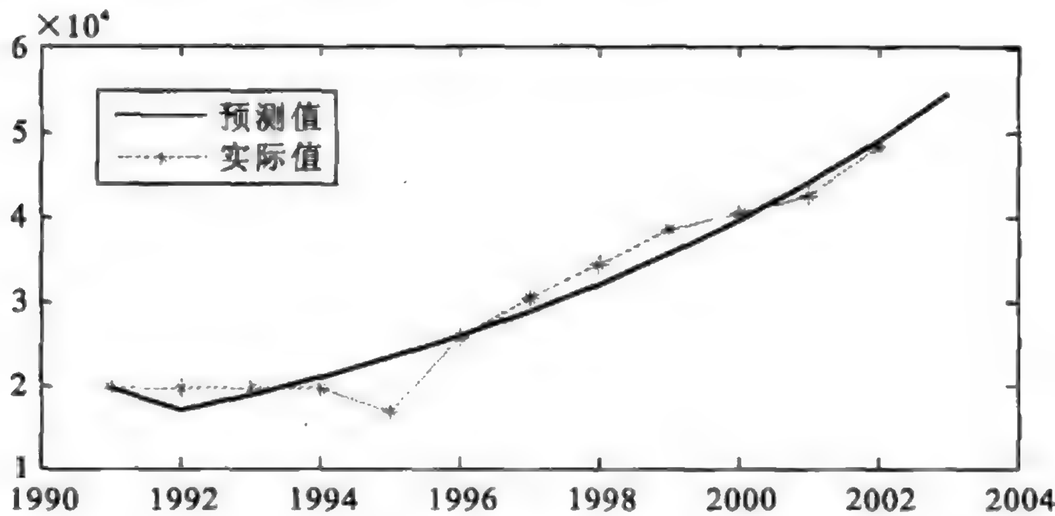


图 9.7 1990～2002 年 GM(1,1)灰色系统预测值与实际值比较

结果分析及模型检验 由表 9.9 与图 9.7 可以看出模型的预测值与实际值偏差不大,其后验差检验数 $C=0.16$ 小于 0.35,所以模型精度为优.

9.3.3 方法评论

灰色系统具有能够利用“少数据”建模寻求现实规律的良好特性,克服了资料不足或系统周期短的矛盾.

另外,如果需要做中长期预测时,可以利用名为“新陈代谢灰色系统”的改进模型进行预测. 由于篇幅有限,这里不作详细介绍.

9.4　层次分析法

在现实世界中，往往会遇到决策的问题，如何选择旅游景点的问题，选择升学志愿的问题等. 在决策者作出最后的决定以前，必须考虑很多方面的因素或者判断准则，最终通过这些准则作出选择. 例如，选择一个旅游景点时，可以从宁波、普陀山、浙西大峡谷、雁荡山和楠溪江中选择一个作为自己的旅游目的地，在进行选择时，所考虑的因素有旅游的费用、旅游地的景色、景点的居住条件和饮食状况以及交通状况等. 这些因素是相互制约、相互影响的. 我们将这样的复杂系统称为一个决策系统. 这些决策系统中很多因素之间的比较往往无法用定量的方式描述，此时需要将半定性、半定量的问题转化为定量计算问题. 层次分析法是解决这类问题行之有效的方法. 层次分析法将复杂的决策系统层次化，通过逐层比较各种关联因素的重要性来为分析、决策提供定量的依据.

层次分析法(analytic hierarchy process，简称 AHP)是将决策总是有关的元素分解成目标、准则、方案等层次，在此基础之上进行定性和定量分析的决策方法. 该方法是美国运筹学家匹茨堡大学教授萨蒂于 20 世纪 70 年代初，在为美国国防部研究"根据各个工业部门对国家福利的贡献大小而进行电力分配"课题时，应用网络系统理论和多目标综合评价方法，提出的一种层次权重决策分析方法. 这种方法的特点是在对复杂的决策问题的本质、影响因素及其内在关系等进行深入分析的基础上，利用较少的定量信息使决策的思维过程数学化，从而为多目标、多准则或无结构特性的复杂决策问题提供简便的决策方法. 尤其适合于对决策结果难于直接准确计量的场合.

9.4.1　理论介绍

层次分析法的基本步骤如下：

(1) 建立层次分析结构模型. 深入分析实际问题，将有关因素自上而下分层(目标-准则或指标-方案或对象)，上层受下层影响，而层内各因素基本上相对独立.

(2) 构造成对比较阵. 用成对比较法和 1～9 尺度，构造各层对上一层每一因素的成对比较阵.

(3) 计算权向量并作一致性检验. 对每一成对比较阵，计算最大特征根和特征向量作一致性检验，若通过，则特征向量为权向量.

(4) 计算组合权向量作组合一致性检验，组合权向量可作为决策的定量依据.

下面我们以一个选择旅游境地为例对层次分析法的各个步骤进行详细讲解.

9.4.2 案例分析

例 9.11 可选择旅游地为桂林、黄山、北戴河. 问如何在三个目的地中按照景色、费用、居住、饮食、旅途条件因素进行选择.

(1) 建立层次分析模型,如图 9.8 所示.

目标层:选择旅游境地;

准则层:景色、费用、居住、饮食、旅途;

方案层:桂林、黄山、北戴河.

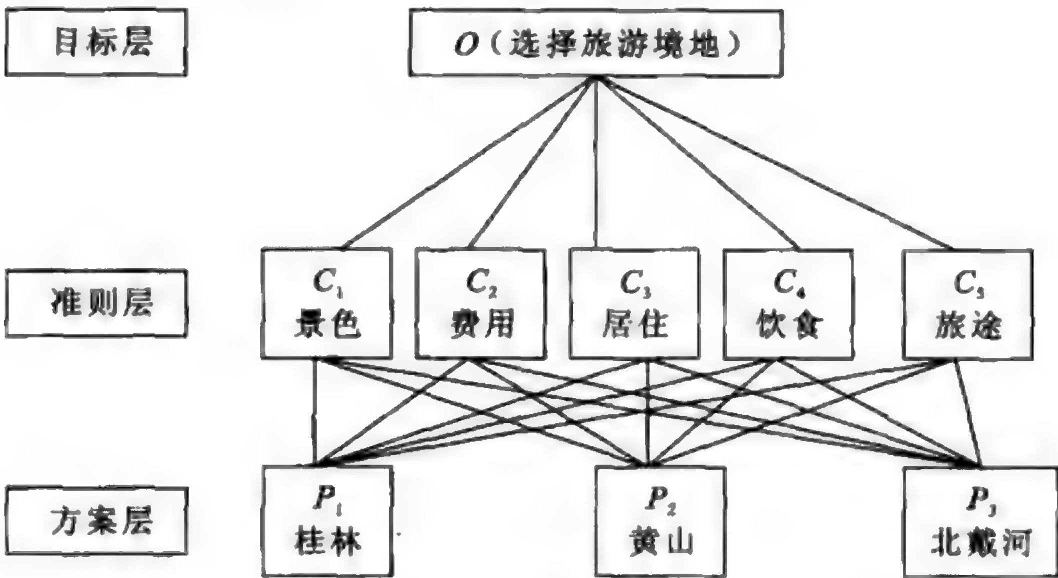


图 9.8 选择旅游地的层次分析模型

(2)构造成对比较阵. 元素之间两两对比,对比采用相对尺度. 设要比较各准则 $C_1,C_2,\cdots,C_n$ 对目标 O 的重要性.

$$C_i:C_j\Rightarrow a_{ij}$$

$$\mathbf{A}=(a_{ij})_{n\times n},\quad a_{ij}>0,a_{ji}=\frac{1}{a_{ij}}$$

尺度采用 Saaty 等人提出 1～9 尺度——a_{ij} 取值 1,2,…,9 及其倒数 1,1/2,…,1/9,便于定性到定量的转化,见表 9.10.

表 9.10 相对尺度参考表

尺度 a_{ij}	1	2	3	4	5	6	7	8	9
$C_i:C_j$的重要性	相同		稍强		强		明显强		绝对强

心理学家认为成对比较的因素不宜超过9个. 用 1～3,1～5,…,1～17,…,$1p$～$9p(p=2,3,4,5)$,$d+0.1$～$d+0.9$ $(d=1,2,3,4)$等 27 种比较尺度对若干实例构造成对比较阵,算出权向量,与实际对比发现,1～9 尺度较优. 得出准则层对于目标的对比阵矩阵:

$$
\mathbf{A}=\begin{bmatrix} 1 & 1/2 & 4 & 3 & 3 \\ 2 & 1 & 7 & 5 & 5 \\ 1/4 & 1/7 & 1 & 1/2 & 1/3 \\ 1/3 & 1/5 & 2 & 1 & 1 \\ 1/3 & 1/5 & 3 & 1 & 1 \end{bmatrix}
$$

方案层对准则层的对比阵同样采取上述方法，不再详细说明.

(3) 计算权向量并作一致性检验. 成对比较完全一致的情况：满足 $a_{ij}\cdot a_{jk}=a_{ik}$，$i,j,k=1,2,\cdots,n$ 的正反阵 $\mathbf{A}$ 称一致阵，如

$$
\mathbf{A}=\begin{bmatrix} \frac{w_1}{w_1} & \frac{w_1}{w_2} & \cdots & \frac{w_1}{w_n} \\ \frac{w_2}{w_1} & \frac{w_2}{w_2} & \cdots & \frac{w_2}{w_n} \\ \vdots & \vdots & & \vdots \\ \frac{w_n}{w_1} & \frac{w_n}{w_2} & \cdots & \frac{w_n}{w_n} \end{bmatrix}
$$

一致阵的性质：①A 的秩为 1，A 的唯一非零特征根为 n；②A 的任一列向量是对应于 n 的特征向量；③A 的归一化特征向量可作为权向量.

对于不一致(但在允许范围内)的成对比较阵 $\mathbf{A}$，建议用对应于最大特征根 λ 的特征向量作为权向量 $\boldsymbol{w}$，即 $\mathbf{A}\boldsymbol{w}=\lambda\boldsymbol{w}$.

一致性检验(对 $\mathbf{A}$ 确定不一致的允许范围)：已知 n 阶一致阵的唯一非零特征根为 n，可证 n 阶正互反阵最大特征根 $\lambda\geqslant n$，且 $\lambda=n$ 时为一致阵.

定义一致性指标：$\mathrm{CI}=\frac{\lambda-n}{n-1}$，CI 越大，不一致越严重.

为衡量 CI 的大小，引入随机一致性指标 RI——随机模拟得到 a_{ij}，形成 $\mathbf{A}$，计算 CI 即得 RI.

Saaty 的结果见表 9.11.

表 9.11　随机一致性指标

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

定义一致性比率 CR=CI/RI：当 CR<0.1 时，通过一致性检验. 选择旅游境地中准则层对目标层的权向量及一致性检验. 最大特征根 $\lambda=5.073$，权向量(特征向量) $\boldsymbol{w}=(0.263,0.475,0.055,0.090,0.110)^{\mathrm{T}}$，一致性指标 $\mathrm{CI}=\frac{5.073-5}{5-1}=0.018$，随机一致性指标 RI=1.12(查表)，一致性比率 CR=0.018/1.12=0.016<0.1，通过一致性检验.

第三层对第二层的计算结果见表 9.12.

表 9.12　第三层对第二层的计算结果

k	1	2	3	4	5
	0.595	0.082	0.429	0.633	0.166
$w^{(3)}{}_k$	0.166	0.277	0.236	0.429	0.193
	0.668	0.129	0.682	0.142	0.175
λ_k	3.005	3.002	3	3.009	3
CI_k	0.003	0.001	0	0.005	0

$\mathrm{RI}=0.58(n=3)$，CI_k 均可通过一致性检验. $w^{(3)}{}_k$ 方案层对第 k 个准则的权重向量.

(4) 计算组合权向量(作组合一致性检验). 设 $ww=(w_1^{(3)},\cdots,w_5^{(3)})$，则方案层对目标层的组合权重向量为 $ww*w$，即 $(0.300,0.246,0.456)^{\mathrm{T}}$. 由此可知，应当选择去北戴河旅游.

选择旅游景点问题求解 MATLAB 程序如下：

```
clc
a=[1,1,1,4,1,1/2
   1,1,2,4,1,1/2
   1,1/2,1,5,3,1/2
   1/4,1/4,1/5,1,1/3,1/3
   1,1,1/3,3,1,1
   2,2,2,3,3,1];  %一致矩阵
[x,y]=eig(a);eigenvalue=diag(y);lamda=max(eigenvalue);
ci1=(lamda-6)/5;cr1=ci1/1.24
w1=x(:,1)/sum(x(:,1))
b1=[1,1/4,1/2;4,1,3;2,1/3,1];
[x,y]=eig(b1);eigenvalue=diag(y);lamda=eigenvalue(1);
ci21=(lamda-3)/2;cr21=ci21/0.58
w21=x(:,1)/sum(x(:,1))
b2=[1  1/4  1/5;4  1  1/2;5  2  1];
[x,y]=eig(b2);eigenvalue=diag(y);lamda=eigenvalue(1);
ci22=(lamda-3)/2;cr22=ci22/0.58
w22=x(:,1)/sum(x(:,1))
b3=[1  3  1/3;1/3  1  1/7;3  7  1];
[x,y]=eig(b3);eigenvalue=diag(y);lamda=eigenvalue(1);
ci23=(lamda-3)/2;cr23=ci23/0.58
w23=x(:,1)/sum(x(:,1))
```

```
b4=[1  1/3  5;3  1  7;1/5  1/7  1];
[x,y]=eig(b4);eigenvalue=diag(y);lamda=eigenvalue(1);
ci24=(lamda-3)/2;cr24=ci24/0.58
w24=x(:,1)/sum(x(:,1))
b5=[1  1  7;1  1  7;1/7  1/7  1];
[x,y]=eig(b5);eigenvalue=diag(y);lamda=eigenvalue(1);
ci25=(lamda-3)/2;cr25=ci25/0.58
w25=x(:,1)/sum(x(:,1))
b6=[1  7  9;1/7  1  1;1/9  1  1];
[x,y]=eig(b6);eigenvalue=diag(y);lamda=eigenvalue(1);
ci26=(lamda-3)/2;cr26=ci26/0.58
w26=x(:,1)/sum(x(:,1))
w_sum=[w21,w22,w23,w24,w25,w26]*w1
ci=[ci21,ci22,ci23,ci24,ci25,ci26];
cr=ci*w1/sum(0.58*w1)
```

9.4.3 方法评论

层次分析法路清晰、方法简便、适用面广、系统性强等特点，便于普及推广，可成为人们工作和生活中思考问题、解决问题的一种方法. 将 AHP 引入决策，是决策科学化的一大进步. 它最适宜于解决那些难以完全用定量方法进行分析的决策问题，因此，它是复杂的社会经济系统实现科学决策的有力工具.



参 考 文 献

董麓.2001.数据分析方法.大连:东北财经大学出版社
高尚,杨静宇.2006.群智能算法及其应用.北京:中国水利水电出版社
黄燕,吴平.2006.SAS统计分析及应用.北京:机械工业出版社
黄友锐.2008.智能优化算法及其应用.北京:国防工业出版社
姜启源,谢金星,叶俊.2008.数学模型.第3版.北京:高等教育出版社
姜启源,刑文训,谢金星,杨顶辉.2005.大学数学实验.北京:清华大学出版社
李继成,戴永红.2003.数学实验.西安:西安交通大学出版社
李新蕊.2007.主成分分析、因子分析、聚类分析的比较与应用.山东教育学报,6
刘承平.2002.数学建模方法.北京:高等教育出版社
刘勇,康立山,陈毓屏等.1995.非数值并行计算——遗传算法.北京:人民邮电出版社
曲庆云,赵晓梅,阮桂海等.2004.统计分析方法.北京:清华大学出版社
苏金明,王永利.2007.MATLAB 7.0实用指南(上册).北京:电子工业出版社
汪晓银.2004.中国蔬菜生产、消费与贸易研究——一个供需平衡的计量经济分析框架.华中农业大学硕士论文
汪晓银,邹庭荣.2008.数学软件与数学实验.北京:科学出版社
王燕.2005.应用时间序列分析.北京:中国人民大学出版社
吴启迪,汪镭.2006.群能蚁群算法及应用.上海:上海科技教育出版社
肖枝洪,郭明月.2009.时间序列分析与SAS应用.武汉:武汉大学出版社
谢季坚,刘承平.2006.模糊数学方法及其应用.第3版.武汉:华中科技大学出版社
谢金星.2006.数学模型.第2版.北京:清华大学出版社
谢金星,谢毅.2005.优化建模与LINDO/LINGO软件.北京:清华大学出版社
余家林,肖枝洪.2008.多元统计及SAS应用.武汉:武汉大学出版社
袁新生,邵大宏,郁时炼.2007.LINGO和Excel在数学建模中的应用.北京:科学出版社
于秀林,任雪松.1999.多元统计分析.北京:中国统计出版社
赵静,但琦.2000.数学建模与数学实验.北京:高等教育出版社
周明,孙树栋.1999.遗传算法原理及其应用.北京:国防工业出版社

(O-3782.0101)

数学建模与数学实验



ISBN 978-7-03-026502-9

定价：28.00 元

[General Information]
书名=数学建模与数学实验
作者=汪晓银主编
页数=268
出版社=北京市：科学出版社
出版日期=2010.01
SS号=12590507
DX号=000006868740
URL=http://book.szdnet.org.cn/bookDetail.jsp?dxNumber=000006868740&d=232A7277B6FCB2D27DF9F42CB07175BD